THE MYSTERIOUS WORLD OF QUANTUM COMPUTING
Advanced Computing and Communications Society (ACCS) is launching India’s first free Open Access digital Library (OPAL) for discovering technical content published in India by recognized technical and scientific conferences and journals.

OPAL provides a robust indexing platform open to all ACCS members (both individuals and corporate members). Members wishing to use OPAL services should register with the platform. Registration is free and requires only minimal documentation to authenticate the authors and/or organizations. OPAL does not own or claim copyright to materials indexed by it or hosted by it, unless explicitly specified.

Salient Features

1. Search: Search based on the metadata of the technical content
2. Download papers and articles: Download technical content in PDF format
3. Upload: Upload of technical content by authenticated users in a specified format and metadata
4. Diverse content: Text documents, Documents with scientific/mathematical notations, Datasets etc.
5. Omni-Access: Easily accessible on all popular form factors of screens (Desktop/Mobile/Tablet...)
6. API made available: The key features of the OPAL will be exposed to users via an API

Become an Institutional Member of ACCS and start a local ACCS Chapter in your Institute!

An Institutional membership is truly an investment in the future of our engineers. The membership also increases your institution’s visibility in the scientific community and encourages collaborative research.

For details visit www.accsindia.org/membership

Advanced Computing and Communications Society
Gate No. 2, C. V. Raman Avenue
Indian Institute of Science
Bengaluru - 560012, Karnataka, INDIA

- Telephone: +91 80 2360 7142
- Mobile: +91 76249 57142
- Email: info@accsindia.org
- Website: http://www.accsindia.org
A perplexing aspect of quantum mechanics is that it defies our normal intuition. It is dramatically different from classical physics as built by Newton, Maxwell and Einstein.

Demonstrates how GALA approach can help many of our students develop mathematical and computational thinking abilities. They have introduced GALA approach through three examples from number theory and geometry which brings out apparent paradoxes along with explanations which is bound to generate curiosity in students going through DSA course.

Development of a mobile app as a student project that uses truncated Latitude and Longitude co-ordinate representation for specifying location of any point on earth.

On February 15, 1946, ENIAC (Electronic Numerical Integrator and Computer) was formally unveiled at the Moore School of Electrical Engineering of the University of Pennsylvania at Philadelphia in the United States of America. This year marks its 75th anniversary.
The euphoric rise in classical computing speeds over the years is traceable to applicability of Moore's law as regards the shrinking geometrical size of CMOS transistors and usage of the Von Neumann architecture. With the technology in this area almost reaching saturation point, there is an interest to explore an alternative technology like quantum computing to improve computational speeds.

Quantum computing harnesses the phenomena of quantum mechanics to take a huge leap forward in computation to solve certain problems. They are capable of solving complex problems that even today's most powerful supercomputers cannot solve, and never will.

As an example, suppose we need to search for one item from a list of N items. On a classical computer we would have to check N/2 items on average, and in the worst case we need to check all N.

Using Grover's search on a quantum computer, we would find the item after checking roughly $\sqrt{N}$ of them. This represents a remarkable increase in processing efficiency and time saved. For example, if we wanted to find one item in a list of 1 trillion, and each item took 1 microsecond to check, a classical computer would take about a week while a Quantum computer would take about a second.

In the cover feature article of this issue of ACC, the author takes the reader through a short tutorial on quantum computing and is meant for those who have already begun to study quantum computing, are familiar with the basic terminologies of quantum mechanics and its core concepts (quantum superposition, measurement, entanglement, de-coherence, etc.) and would like to step back and reflect upon the foundations of quantum computing before getting into the nitty gritty of more complex algorithms, e.g., Shor's factorization algorithm, Grover's search algorithm, etc.

Moving on to the next article, titled 'A Gamified Approach to Learning Algorithms (GALA)', the authors take the readers through some fascinating problems encountered in Data Structures and Algorithms (DSA) course which is an important and a core course in an undergraduate degree curriculum in computer science branch of engineering. Their article demonstrates how GALA approach can help many of our students develop mathematical and computational thinking abilities. They have introduced GALA approach through three examples from number theory and geometry which brings out apparent paradoxes along with explanations which is bound to generate curiosity in students going through DSA course.

In the feature article titled 'A Lat - Long code', the author proposes development of a mobile app as a student project that uses truncated Latitude and Longitude co-ordinate representation for specifying location of any point on earth. This representation uses 8 digits for Latitude representation (Sxx.xxxx) and 9 digits for Longitude representation (Syyy.yyyy) with S taking on a + sign for a point on earth that is in northern hemisphere and - sign for a point in southern sphere. It is claimed that given two points described by proposed Lat-Long representation, it is easy to perceive not only the direction of one point with respect to other but also easy to estimate the approximate distance of one from the other manually. Such a computation is expected to result in computation of distance with less than 15 meters of accuracy. Also do not miss the historical account of building ENIAC on its 75th anniversary, chronicled by none other than Dr. V. Rajaraman.

Dr. N Rama Murthy
Editor
THE MYSTERIOUS WORLD OF QUANTUM COMPUTING

RAJENDRA K. BERA
Chief Mentor, Acadinnet Education Services India Pvt. Ltd., Bangalore

https://doi.org/10.34048/2021.2.F1
Abstract
This paper is intended to be a short tutorial for those who have already begun to study quantum computing, are familiar with the basic terminologies of quantum mechanics and its core concepts (quantum superposition, measurement, entanglement, decoherence, etc.) and would like to step back and reflect upon the foundations of quantum computing before getting into the nitty gritty of more complex algorithms, e.g., Shor’s factorization algorithm, Grover’s search algorithm, etc. The important thing to remember is that while the mathematics used to deal with classical and quantum physics is the same, interpreting quantum measurement requires a twist. That interpretation is governed by a separate postulate that tells you the probability with which you will see various parts of the answer in a given experiment. This complication does not exist in classical physics. Once you come to terms with this postulate, you will find quantum computing rather easy to deal with. Measurement is a complex matter and necessitates that the experiment be repeated many times and measurements accumulated before analysis of the results is made to reveal the full answer. The repetitions can be done in a series, in parallel or in any combination of the two. The experiments must be identical in their mathematical description and input data.

Keywords: Quantum, qubit, measurement, superposition, entanglement, algorithm, gate.

1 Introduction
A perplexing aspect of quantum mechanics is that it defies our normal intuition. It is dramatically different from classical physics as built by Newton, Maxwell, and Einstein. Laws of classical physics are deterministic in the sense that given, say, Newton’s laws of motion, and initial conditions (position and momentum) at some instant we deem to be \( t = 0 \) for a system and a time history of the force(s) acting on the system, we can, in principle, accurately predict the state of that system at any time in the past or the future. Further, we can measure the present state of the system (position and momentum) without disturbing it.

In quantum mechanics, the situation is completely different. The counterpart of Newton’s second law of motion for a quantum system is the Schrödinger’s wave equation, and the state of the system is described by something called the “wave function”, \( |\psi\rangle \), which no one understands intuitively. It is so abstract that we understand it only in a mathematical sense. Newton’s first law of motion refers to ‘force-free’ motion.

It has not been possible for physicists (or anyone else for that matter) to understand the wave function in any other way. If we try to measure the state of a quantum system, hell breaks loose; we have no way of deterministically predicting what the result of a measurement will be! And, even in principle, there is no way we can measure a quantum system without disturbing it. That is why physics is divided into two parts: classical physics, and quantum physics.\(^1\)

No one knows what transitory changes a quantum system undergoes when it is measured. We do know, however, that while we cannot deterministically anticipate the result of a measurement, we can make an amazingly accurate probabilistic prediction of it.\(^2\) And, even in principle, there is no way we can measure a quantum system without disturbing it.

\[^{1}\text{See, e.g., Bera (2018a, b).}\]
\[^{2}\text{Bera & Menon (2009). See also: Bera (2020), Chapter 12.}\]
\[^{3}\text{Born (1926).}\]
\[^{4}\text{Born (1954). Strongly recommended for reading.}\]
2 Axioms of Quantum Mechanics

Here are the laws (or postulates or axioms) of quantum mechanics stated informally.

- Quantum mechanics describes a physical system through a mathematical object called the state vector (or the wave-function) $|\psi\rangle$, which is complex (i.e., it has real and imaginary parts) and it is a vector of unit length. (Our social systems too are complex because humans have real and imaginary aspirations!)

- $|\psi\rangle$ evolves in a deterministic manner according to the linear Schrödinger equation:

$$-\frac{\hbar^2}{2m}\nabla^2|\psi(r, t)\rangle + V(r)|\psi(r, t)\rangle = i\hbar\frac{\partial|\psi(r, t)\rangle}{\partial t},$$

where $|\psi\rangle$ is the wavefunction, $\hbar$ is the reduced Planck's constant, $V$ is the potential energy of the particle at position $r$ (it assumes the presence of conservative forces only), $m$ is the mass of the particle under consideration, the operators $\nabla^2$ and $\frac{\partial}{\partial t}$, respectively, describe how the wavefunction changes with space and time. In particular, bear in mind that $|\psi\rangle$ remains a unit vector during its evolution, and any operation on it only changes its orientation.

- Any measurement made on a quantum system leads to the (non-unitary) irreversible collapse of its wave function to a new state governed by a probability rule prescribed by Born. Measurement destroys quantum superposition.

- The state space of a composite quantum system is the tensor product of the state spaces of the component quantum systems.

One cannot overemphasize the fact that it is only when measurements are made that indetermination and probabilities come into quantum theory. Otherwise, things are very deterministic. The collapse of the wave function involves no forces of any kind but it does involve loss of information, hence a measurement cannot be undone. (In classical physics, a measurement is deemed possible without changing the state of the system.)

2.1 An Example of What the Axioms Mean

Consider the simple case of a quantum system capable of being in two distinctly different states $|F\rangle$ and $|G\rangle$. The general state of the wavefunction $|\psi\rangle$ is then given by the linear combination of $|F\rangle$ and $|G\rangle$ as

$$|\psi\rangle = a |F\rangle + b |G\rangle,$$

where $a$ and $b$ are complex constants (with real an imaginary parts). When $|\psi\rangle$ is measured, the output will not be

$$a |F\rangle + b |G\rangle.$$

It will be either

$$|\psi\rangle = |F\rangle, \text{ or } |\psi\rangle = |G\rangle.$$

The probability (prob) with which one or the other will appear in the output is as follows:

$$\text{prob} |F\rangle : \text{prob} |G\rangle = |a|^2 : |b|^2 ;$$

$$|a|^2 + |b|^2 = 1.$$

But this will become visible only if a very large number of identical $|\psi\rangle$ systems are available or can be created and a measurement on each system is made. When measured, each system will collapse to either $|F\rangle$ or $|G\rangle$ in an unpredictable way in proportion to the probabilities noted above and will remain in that state if left undisturbed. It will not be in any linear combination of $|F\rangle$ and $|G\rangle$.

The collapse of the wave function seems to happen instantly unlike the ordinary time evolution of quantum states (according to the Schrödinger's equation). We still do not understand the physical mechanism, which causes the collapse. Complex linear superposition of states and collapse of wave functions are unusual features of quantum mechanics. If you get infatuated with quantum mechanics, you will eventually fall in love with it.

When $|\psi\rangle$ is manipulated by any operator, it remains a unit vector, e.g., when $|\psi\rangle = |F\rangle$ or $|\psi\rangle = |G\rangle$ after measurement; only $a$ and $b$ change such that $|a|^2 + |b|^2 = 1$. All of quantum computing is about manipulating complex constants.

A simple generalization exists when the wavefunction $|\psi\rangle$ has more than two, say, $n$, distinctly different states $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle$. Its linearly superposed state is then described by

$$|\psi\rangle = a_1 |\psi_1\rangle + a_2 |\psi_2\rangle + \ldots + a_n |\psi_n\rangle,$$

where $a_1, a_2, \ldots, a_n$ are complex constants, which may change if $|\psi\rangle$ is operated upon by either quantum operators or measurement operators. After a measurement,
i.e., the application of a measurement operator, $|\psi\rangle$ will collapse to $|\psi_i\rangle$ with the index $i$ occurring with certain probabilities as noted below:

$$\text{prob} |\psi_1\rangle : \text{prob} |\psi_2\rangle : \ldots : \text{prob} |\psi_n\rangle = |a_1|^2 : |a_2|^2 : \ldots : |a_n|^2 ;$$

$$|a_1|^2 + |a_2|^2 + \ldots + |a_n|^2 = 1.$$

Thus, when $|\psi\rangle$ changes either due to Schrödinger evolution or measurement, only the $a_i$ change such that the sum of the $|a_i|^2$ always remains 1. This ensures that $|\psi\rangle$ will remain a unit vector and only its orientation will change.

### 2.2 An Example of Output Weirdness

At first glance, what did you see?

$$|\psi\rangle = a|F\rangle + b|G\rangle$$

Take a quick glance at Figure 1 and note in your mind what you saw. Now look at the figure again moving your eyes up and down. Did you see another picture? Did you see anything other than sometime a beautiful girl (the wife of the artist) and another time an old woman (his mother-in-law). You did! More to the point, you did not see any weird combination of your two visual interpretations. It is the same picture (by analogy the wavefunction $|\psi\rangle$) which is a combination of the fair lady $|F\rangle$ and the grand lady $|G\rangle$. When viewed, you were able to see only one of them at any time.

What is not captured in Figure 1 is that whatever you saw did not get frozen. In a quantum measurement, the wavefunction freezes (or collapses) to the measured value. But what the picture does capture is that a given matrix of pixels can carry more than one meaning in the same space and at the same time. This is a quintessential wave property. Two pieces of matter cannot occupy the same space at the same time, waves and interpretations can.

Hopefully, you now have developed some gut feeling for what one means by superposition and wavefunction collapse in quantum mechanics. By the way, during the interval when your eyes first fell on the picture till the moment you “saw” the picture, was your brain in a state of superposition? Were you in two minds simultaneously? Is the wavefunction aware of being in a state of superposition? These are questions you might wish to explore further, perhaps philosophically.

### 3 Unitary Transformation in Quantum Mechanics

The Schrödinger evolution of the wave function, alternatively, can be described in matrix form due to Werner Heisenberg, which is the form used in quantum computing. In this form, the evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi(t_1)\rangle$ of the system at time $t_1$ is related to the state $|\psi(t_2)\rangle$ of the system at time $t_2$ by a unitary operator $U$ which depends only on the times $t_1$ and $t_2$.

$$|\psi(t_2)\rangle = U|\psi(t_1)\rangle.$$ 

A linear operator $U$ whose inverse is its adjoint (conjugate transpose, $U^\dagger$) is called unitary, that is, $U^\dagger U = I$.
$UU^\dagger = I$, where $U^\dagger \equiv (U^*)^T \equiv (U^T)^*$. Thus, by definition, unitary operators are invertible. Also, by definition, a unitary operator does not change the length of the state vector it acts upon; it only changes that vector’s orientation. This means that if

$$|\psi(t_1)\rangle = a_1|\psi_\alpha\rangle + b_1|\psi_\beta\rangle; \quad |a_1|^2 + |b_1|^2 = 1,$$

then

$$|\psi(t_2)\rangle = a_2|\psi_\alpha\rangle + b_2|\psi_\beta\rangle; \quad |a_2|^2 + |b_2|^2 = 1.$$

4 Quantum Computing

Quantum computing is about computing with quantum systems using the rules of quantum mechanics rather than the rules of classical mechanics. The important quantum mechanical phenomena that come into play in the building of a quantum computer are:

- Superposition
- Entanglement
- Decoherence

On a quantum computer, problem specific algorithms are executed that comprise an organized sequence of unitary operations and measurements once relevant input data is provided. Since all unitary operators are invertible, we can always reverse or ‘uncompute’ those computations. (In principle, we can do this on Turing machines using classical physics also.\(^5\)) If a measurement is made, ‘uncomputing’ previous unitary operations are no longer possible. That is, measurements cannot be undone because it is axiomatically tied to the probabilistic nature of quantum measurement and collapse of the wave function.

4.1 Quantum Superposition

Let us look at Figure 1 “My wife, and my mother-in-law” once again. You cannot really define a linear combination (superposition) of

$$|\psi\rangle = a|F\rangle + b|G\rangle = a|\text{my wife}\rangle + b|\text{my mother—in-law}\rangle$$

in a physical sense, yet you know that in some “complex” sense the two people are superposed in the picture (i.e., they exist simultaneously at the same place and time). You recognize the superposition at an intellectual level, but not at the measurement level (vision); you see the picture “collapsing” to one or the other person. In quantum mechanics, the measurement operator is like a prism—it splits the wavefunction into its component parts (akin to white light being split into its rainbow components by a prism). In our example, the “prism” would split the picture into |my wife) and |my mother—in-law) and the probability with which we will see one or the other is given by

$$\text{prob}|F\rangle : \text{prob}|G\rangle = \text{prob}|\text{my wife}\rangle : \text{prob}|\text{my mother—in-law}\rangle = |a|^2 : |b|^2 = 1.$$

You will not see some weird hybrid form of “my wife” and “my mother—in-law”.

4.2 Quantum Entanglement

Entanglement is a form of quantum superposition. It is a joint and acquired group property of two or more quantum particles, which when entangled, share an instantaneously responsive bonded existence even when light years apart. Indeed, separation distance is irrelevant. If the state of one is changed, the state of the other is instantaneously adjusted to be quantum consistent. If a measurement is made on one, both will collapse together. There is no easy explanation of entangled correlations. There is no counterpart of entanglement in classical mechanics.

Einstein called such action-at-a-distance ‘spooky’. He wrote to Max Born in March 1947:

> I cannot seriously believe in [the quantum theory] because it cannot be reconciled with the idea that physics should represent a reality in time and space, free from spooky actions at a distance.\(^6\)

4.3 Decoherence

It is the spontaneous disruptive interaction between a quantum system and its environment which destroys quantum superposition. The reason why quantum computers still have not replaced our laptops is that superposition and entanglement are extremely fragile states. Any interaction with the environment and the particles tend to decohere. Preventing decoherence from taking hold before a calculation is completed remains the biggest challenge in building quantum computers. While considerable progress has been made in dealing with this problem, still more needs to be done. (See also Sec. 10 below.)

\(^5\) Bennett (1982).

\(^6\) Quote taken from Corliss (1997).
5 Physical Laws are Mathematical

When we say $F = ma$ expresses Newton's second law of motion, what we mean is that if you interpret $F$ as representing a force (a vector with 3-scalar components), $m$ representing the mass of a material body, and a representing the acceleration of that material body, then we can very accurately compute the motion of that material body. $F = ma$ means nothing until we give it an interpretation. Surprisingly, all the important laws of physics can be precisely stated in mathematical form. This fact led the 1963 Nobel Laureate in physics, Eugene Paul Wigner, to comment in wonder, The Unreasonable Effectiveness of Mathematics in the Natural Sciences. It turns out that without knowing mathematics, you cannot develop a deep understanding of physics. This is particularly true for quantum mechanics. Warning: Unless you understand linear algebra and complex variable theory, you will not understand quantum mechanics.

5.1 Interpretation of Schrödinger’s Equation

Recall Schrödinger’s equation

$$\frac{\hbar^2}{2m} \nabla^2 |\psi(r, t)\rangle + V(r) |\psi(r, t)\rangle = i \hbar \frac{\partial |\psi(r, t)\rangle}{\partial t}.$$ 

Even though we do not know what the wavefunction $|\psi\rangle$ means, yet, with the following interpretation:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Classical → quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time, $t$</td>
<td>$t \rightarrow t$</td>
</tr>
<tr>
<td>Position, $r$</td>
<td>$r \rightarrow r$</td>
</tr>
<tr>
<td>Momentum, $p$</td>
<td>$p \rightarrow -i\hbar \nabla$</td>
</tr>
<tr>
<td>Energy, $E$</td>
<td>$E \rightarrow i\hbar \left( \frac{\partial}{\partial t} \right)$</td>
</tr>
</tbody>
</table>

We make a connection with the physical world. Note that we are interpreting two mathematical operators as physical variables: the operator $-i\hbar \nabla$ is interpreted to represent the physical variable momentum $p$ and the operator $i\hbar \left( \frac{\partial}{\partial t} \right)$ as the physical variable energy $E$. Now you can understand why quantum mechanics appears weird. It requires a lot of imagination to connect the real world with complex number mathematics.

Why are quantum systems quantized?

Let us call $\hat{H}$ the Hamiltonian operator where

$$\hat{H} \equiv -\frac{\hbar^2}{2m} \nabla^2 + V,$$

Then from the table above

$$\hat{H} |\psi\rangle = E |\psi\rangle.$$ 

Since the Hamiltonian operator can alternatively be written as a square matrix, we have an equation of the form we call an eigenvalue problem in linear algebra:

$$Ax = \lambda x.$$ 

This equation has the trivial solution, $x = 0$, but it also has non-trivial solutions for certain discrete values of $\lambda$ (which we call the eigenvalues of matrix $A$). So now we see why, for a finite quantum system, energy is quantized. The $E$ values are the eigenvalues of the Hamiltonian operator! A quantum system is quantized because the Schrödinger’s equation describes an eigenvalue problem.

5.2 Heisenberg’s Uncertainty Principle

We now find another intriguing aspect of quantum mechanics. This comes from the fact that operators, do not always commute. Thus, $pr \neq rp$, because

$$-i\hbar \nabla r \neq r \left( -i \hbar \nabla \right).$$

This means that the measurement of position $r$ and momentum $p$ of a particle is no longer independent of the sequence in which they are measured. This is the reason why one cannot measure both position and momentum of a quantum particle with absolute accuracy. And this fact is responsible for the famous Heisenberg’s uncertainty principle in quantum mechanics, which says:

$$\Delta p \Delta q \geq \frac{\hbar}{2},$$

where $\Delta q$ is the error in the measurement of any coordinate and $\Delta p$ is the error in its canonically conjugate momentum. In quantum mechanics, e.g., position ($q$) and momentum ($p$) of a particle are such measurements. The uncertainty principle is categorical. We cannot observe

\[Wigner (1960).\]
a quantum system without affecting it. The independent, unobtrusive observer simply does not exist. Thus, contrary to Newtonian mechanics, even in principle, it is not possible to know enough about the present to make a complete prediction about the future. The measurement limits in quantum mechanics cannot be overcome by refining measurement technology as we penetrate into the subatomic world. Classical and quantum particles are entirely different entities. The principle sets limits on precision technologies, e.g., metrology and lithography. However, if the particle is prepared entangled with a quantum memory (such as an optical delay line) and the observer has access to the particle stored in the quantum memory, it is possible to predict the outcomes for both measurement choices precisely. This is a more general uncertainty relation, formulated in terms of entropies. This new relation has been verified experimentally.

5.3 No Cloning, No Deletion
That quantum operators are unitary, presents unusual consequences. One is that if you do not know the state of a quantum system (even if it is a single particle) then you cannot make an exact copy of it. This is known as the no-cloning theorem. (You can, of course, prepare a particle in any desired state and make as many copies of it as you like.) The other is that unless a quantum system collapses, you cannot delete information in a quantum system. This is known as the no-deletion theorem. These results are connected with the quantum phenomenon called entanglement. Quantum algorithm designers make very clever use of quantum superposition and quantum entanglement.

5.4 Ingredients of Quantum Algorithms
In the weird quantum world, quantum algorithms are replete with clever sequences of unitary operators that move the system according to Schrödinger’s equation, of measurement operators that collapse the system, and very imaginative uses of superposition and entanglement.

A single qubit (the quantum analogue of the classical bit) is the simplest quantum system we can think of. Mathematically, a qubit is described as a unit vector \( |\psi\rangle = a|0\rangle + b|1\rangle \), parameterized by two complex numbers \( a \) and \( b \), satisfying \( |a|^2 + |b|^2 = 1 \). While the qubit can be in either state \( |0\rangle \) or \( |1\rangle \) (analogous to the 0 and 1 states of a classical bit), it can also be in a superposed state of \( (a|0\rangle + b|1\rangle) \), which a classical bit can never be in. In any non-trivial computation, of course, many more than one qubit will be required.

6 Manipulating Qubits
In quantum computing we rely on the following facts:

(1) All quantum gates are reversible by design, i.e., unitary.

(2) Any unitary operation on \( n \) qubits can be implemented exactly by stringing together operations composed of 1-qubit Pauli operators and 2-qubit controlled-NOT gates. This is provable. These gates are described below.

6.1 1-qubit Unitary Operators
Any unitary operator \( M \) changing the state of a single qubit can always be set up as a linear combination of 4 unitary operators (also called gates), traditionally represented by \((I, X, Y, Z) \equiv (I, \sigma_0, \sigma_1, \sigma_2, \sigma_3)\), i.e.,

\[
M = \alpha I + \beta X + \gamma Y + \delta Z,
\]

where \( \alpha, \beta, \gamma, \delta \) are complex constants. The operators \((I, X, Y, Z)\) are called Pauli matrices, and in their alternative symbolic form \((\sigma_0, \sigma_1, \sigma_2, \sigma_3)\) are called sigma matrices. Each operator is a 2x2 matrix. The Hadamard gate (a linear combination of \( X \) and \( Z \) gates) is a very important and often used gate.

---

8Berta (2010).
Hadamard gate, $H$

The Hadamard gate, $H \equiv (X + Z)/\sqrt{2}$, is an amazing gate. When applied to a qubit in state $|0\rangle$ or $|1\rangle$, it changes the qubit’s state, respectively to

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{and} \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$  

When a measurement is now made on the qubit, the output in either case will be $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$ with equal probability. Thus, quantum physics provides us with a perfect random number generator (something impossible in classical physics). In general, if $|\psi\rangle = a|0\rangle + b|1\rangle$, then

$$H|\psi\rangle = a\frac{|0\rangle + |1\rangle}{\sqrt{2}} + b\frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$  

However, if no measurement is made, and the Hadamard gate is applied once again to the output, then

$$H(H|0\rangle) = H\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0\rangle \quad \text{and} \quad H(H|1\rangle) = H\frac{|0\rangle - |1\rangle}{\sqrt{2}} = |1\rangle,$$

and we get back the original state of the qubit, i.e.,

$$H(H|\psi\rangle) = a|0\rangle + b|1\rangle = |\psi\rangle.$$  

This is rather amazing. A randomizing operation to a random state produces a deterministic outcome! The Hadamard gate can randomize and de-randomize.

$n$-qubit Hadamard gate

When the Hadamard gate is applied to $n$ qubits individually, we get,

$$HH \ldots H \quad \text{n-times} \quad |00\ldots 0\rangle = \frac{1}{\sqrt{2}}(|00\ldots + |11\ldots 1\rangle = \frac{1}{\sqrt{2}}(|00\ldots | + |11\ldots 1\rangle)$$

or in the more compact notation $H^{\otimes n} \equiv HH \ldots H$, we get

$$H^{\otimes n} |00\ldots 0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x} |x\rangle,$$

where the sum is over all possible $2^n$ mutually orthogonal states of $n$ qubits or “values” of $x$. Thus the Hadamard transform produces an equal superposition of all $2^n$ possible computational basis states, i.e., $x$ can be viewed as the binary representation of the numbers from 0 to $(2^n - 1)$, and this is done extremely efficiently (as compared to a classical computer) by using only $n$ gates. The $n$-qubit Hadamard gate is sometimes known as the Walsh-Hadamard gate.

6.2 2-qubit Unitary Controlled-not Operator

The Controlled-NOT operator (gate), $C_{\text{not}}$ is an indispensable operator in quantum computing. It is used to entangle two-qubits. It acts on a qubit-pair such that

$$C_{\text{not}}|00\rangle = |00\rangle, \quad C_{\text{not}}|01\rangle = |01\rangle, \quad C_{\text{not}}|10\rangle = |11\rangle, \quad C_{\text{not}}|11\rangle = |10\rangle$$

Note that the state of the first qubit, called the control qubit, does not change while the state of the second qubit, called the target qubit, changes only if the control qubit is in state $|1\rangle$. This signifies the exclusive-or (XOR) operation (that is, the output is “true” if and only if exactly one of the two operands has a value of “true”).

Entangling qubits

An application of the $H$-gate on the first qubit followed by the $C_{\text{not}}$-gate to a 2-qubit system (with the first qubit as the control qubit) gives the following results when the system’s initial state is $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, respectively:

$$C_{\text{not}}(H|00\rangle) = C_{\text{not}}(|00\rangle + |10\rangle)/\sqrt{2} = (|00\rangle + |11\rangle)/\sqrt{2},$$

$$C_{\text{not}}(H|01\rangle) = C_{\text{not}}(|00\rangle - |11\rangle)/\sqrt{2} = (|01\rangle - |10\rangle)/\sqrt{2},$$

$$C_{\text{not}}(H|10\rangle) = C_{\text{not}}(|00\rangle + |10\rangle)/\sqrt{2} = (|01\rangle + |11\rangle)/\sqrt{2},$$

$$C_{\text{not}}(H|11\rangle) = C_{\text{not}}(|01\rangle - |11\rangle)/\sqrt{2} = (|00\rangle - |10\rangle)/\sqrt{2}.$$  

The final states (last column) are called Bell states (after John Bell).

They are interesting because they are all entangled states, that is, they cannot be attained by manipulating each qubit using the 1-qubit Pauli operators alone. If the states of the entangled particles are used to encode bits, then the entangled joint state represents what is called an ebit (entangled qubit pair). Its joint state is always distributed between two qubits. The states of these qubits are correlated, but undetermined to an observer until measured.

3-qubit Toffoli gate, $T$

The $T$ gate (named after Tommaso Toffoli who invented it) acts on a qubit-triplet such that

$$T|000\rangle = |000\rangle, \quad T|100\rangle = |100\rangle,$$
It can be viewed as a controlled-controlled-NOT gate, which negates the last of three qubits, if and only if the first two are 1. The Toffoli gate is its own inverse. Toffoli gates can be constructed using CNOT gates and several Hadamard gates.

7. Some Elementary Algorithms

7.1 Swapping States Between Two Qubits

New simple examples.

Now apply the $C_\text{not}$ gate to the first two qubits while $I$ leaves the third qubit untouched.

The state of the two qubits can be swapped by applying the $C_\text{not}$ gate as follows in the four possible cases.

The $C_\text{not}$ gate has no effect on $|00\rangle$. An application of the Toffoli gate, $T$, now produces:

$|000\rangle \rightarrow |001\rangle$
$|010\rangle \rightarrow |011\rangle$
$|100\rangle \rightarrow |101\rangle$
$|110\rangle \rightarrow |111\rangle$

The first qubit is the placeholder for $\bar{y}$, the second for $\bar{x}$, and the third for the result of $\bar{x} \cdot \bar{y}$. The first two qubits while $I$ leaves the third qubit untouched.

The $C_\text{not}$ gate has no effect on $|00\rangle$. An application of the Toffoli gate, $T$, now produces:

$|000\rangle \rightarrow |000\rangle$
$|010\rangle \rightarrow |011\rangle$
$|100\rangle \rightarrow |101\rangle$
$|110\rangle \rightarrow |111\rangle$

The first qubit is the placeholder for $\bar{y}$, the second for $\bar{x}$, and the third for the result of $\bar{x} \cdot \bar{y}$. The first two qubits while $I$ leaves the third qubit untouched.

8. Interpretations of Quantum Mechanics

To get a hang of quantum mechanics it is important to understand the many ways in which quantum mechanics is interpreted and just what they mean. Bear in mind that the wave function is an abstract mathematical object. Neither its origin nor its underlying mechanism has been disclosed in the laws of quantum mechanics. In particular, the mechanisms for superposition, entanglement, and measurement have not been elucidated. Hence, they too are open to multiple interpretations. What are found in the various interpretations is that while the formalism of quantum mechanics is widely accepted, there is no single interpretation that everyone finds agreeable.

The disagreements essentially stem from the incompatibility that exists between two evolutionarily independent paths: a quantum system follows the Schrödinger equation, and the 'collapse' mode of measurement.

[on the inequivalence of interpretations]

Non-uniqueness of interpretations
Indeed, without the measurement postulate telling us what we can observe, the equations of quantum mechanics would be just pure mathematics that would have no physical meaning at all. Note also that any interpretation can come only after an investigation of the logical structure of the postulates of quantum mechanics is made. Let me explain what we mean by an interpretation in the context of quantum mechanics.

Form and meaning are separate

For example, Newtonian mechanics does not define the structure of matter. How we interpret or model the structure of matter is largely an issue separate from Newtonian mechanics. However, any model of the structure of matter we propose is expected to be such that it is compatible with Newton’s laws of motion in the realm where Newtonian mechanics rules. If it is not, then Newtonian mechanics as we know it would have to be abandoned or modified or the model of the structure of matter would have to be abandoned or modified. One may also have a partial interpretation and leave the rest in abeyance till further insight strikes us and leads us to a complete or a new interpretation.

A question such as whether a particular result deduced from Newton’s laws of motion is deducible from a given model of material structure is therefore not relevant.

Likewise, as long as an interpretation (or model) of superposition, entanglement, and measurement does not require the axioms of quantum mechanics to be altered, none of the predictions made by quantum mechanics would be incompatible with that interpretation. This assertion is important because in our (Bera-Menon) interpretation we make no comments on the Hamiltonian (in the Schrödinger’s equation), which captures the detailed dynamics of a quantum system. Quantum mechanics does not tell us how to construct the Hamiltonian. In fact, real life problems seeking solutions in quantum mechanics need to be augmented by physical theories that are compatible with the axioms of quantum mechanics. Those axioms provide only the scaffolding around which detailed physical theories are to be built.

We now briefly describe three wildly different interpretations—(1) the Copenhagen view, (2) Bohm’s pilot wave, and (3) Everett’s many worlds.

8.1 Copenhagen View
This is the most popular interpretation used by quantum physicists. In the Copenhagen interpretation (around 1927), one cannot describe a quantum system independently of a measuring apparatus. Indeed, it is meaningless to ask about the state of the system in the absence of a classical measuring system. The role of the observer is central since it is the observer who decides what he wants to measure.

In this interpretation, a particle’s position is essentially meaningless; measurement causes a collapse of the wave function and the collapsed state is randomly picked to be one of the many possibilities allowed for by the system’s wave function; the fundamental objects handled by the equations of quantum mechanics are not actual particles that have an extrinsic reality but “probability waves” that merely have the capability of becoming “real” when an observer makes a measurement. Entanglement is treated as a mysterious phenomenon. The preferred interpretation assumed in this paper is the Copenhagen interpretation.

8.2 Bohm’s Pilot Wave
In Bohm’s interpretation (1952), the whole universe is entangled; its parts cannot be separated. Entanglement is not a mystery; it is mediated by a very special unknown anti-relativistic quantum information field (pilot wave, derivable from Schrödinger’s equation) that does not diminish with distance and that binds the whole universe together. It is an all pervasive field that is instantaneous; it is not physically measurable but manifests itself in terms of non-local (unmediated, instantaneous, and unaffected by the nature of the intervening medium) correlations.

In this interpretation, an electron, e.g., has a well-defined position and momentum at any instant. However, the path an electron follows is guided by the interaction of its own pilot wave with the pilot waves of other entities in the universe.

In fact, Bohm treats measurement as an objective process in which the measuring apparatus and what is observed interact in a well-defined way. At the conclusion of this interaction, the quantum system enters into one of a set of ‘channels’, each of which corresponds to the possible results of the measurement while the other channels become inoperative. In particular, there is no ‘collapse’ of the wave function, yet the wave function behaves as if it had collapsed to one of the channels.14

8.3 Everett’s many worlds
Everett’s interpretation15 is perhaps the most bizarre and yet perhaps the simplest (it is free of the measurement problem because Everett omits the measurement postulate) and, instead, requires us to believe that we inhabit one of an infinite number of parallel worlds!

He assumes that when a quantum system in a given world is faced with a choice, i.e., choosing between alternatives as in a measurement, the system splits into a

14Bohm (1951).
15Everett (1957).
number of systems (worlds) equal to the number of options available. Thus, the world of a qubit in state $|0\rangle + |1\rangle)/\sqrt{2}$ will split into two worlds if the qubit is measured. The two worlds will be identical to each other except for the different option chosen by the qubit—in one it will be in state $|0\rangle$ and in the other it will be in state $|1\rangle$. Each world will also carry its own copy of the observer(s), and each observer copy will see the specific outcome that appears in his respective world. Of course, the worlds can overlap and interact in the overlapping regions. Decoherence, that is, (spontaneous) interactions between a quantum system and its environment will cause the worlds to separate into non-interacting worlds.

9 A Novel View on Measurement

In 2009, I and my student Vikram Menon proposed a sub-Planck-structure for the wave function, and within that structure the meaning of superposition, entanglement, and measurement without affecting the postulates of quantum mechanics. The sub-Planck scale provides us with the freedom to construct mechanisms for our interpretation that are not necessarily bound by the laws of quantum mechanics (just as atomic structure is not bound by Newtonian mechanics). In particular, our interpretation need not even satisfy the Schrödinger wave equation because quantum mechanics is not expected to rule in the sub-Planck scale. The high point of our interpretation is that it is able to explain the measurement postulate as the inability of a classical measuring device to measure at a precisely predefined time.

Superposition

In our interpretation we assume that the sub-Planck scale structure of the wave function is such that the wave function is in only one state at any instant but oscillates between its various “superposed” component states (eigenstates). (There is no expenditure of energy in maintaining the oscillations.) That is, the superposed states appear as time-sliced in a cyclic manner such that the time spent by an eigenstate in a cycle is related to the complex amplitudes $(a, b)$ appearing in the qubit’s wave function, $|\psi\rangle = a|0\rangle + b|1\rangle$. The cycle time $T_c$ of the qubit’s oscillation between states $|0\rangle$ and $|1\rangle$ is much smaller than the Planck time $\approx 10^{-43}$ sec. It is not necessary for us to know the value of $T_c$. We only assert that it is a universal constant. Within a cycle, the time spent by the particle in state $|0\rangle$ is $T_0 = |a|^2 T_c$ and in state $|1\rangle$ is $T_1 = |b|^2 T_c$ so that $T_c = T_0 + T_1$. See Figure 2.

![Figure 2: A time multiplexed view of quantum superposition.](image)

Measurement

Our hypothesized measurement mechanism acts instantaneously (through entanglement) but the instant of actual measurement occurs randomly over a small but finite interval $\hbar t_m$, which is much greater than Planck time (otherwise its actual value is immaterial), from the time the measurement apparatus is activated. In particular, we regard measurement as the joint product of the quantum system and the macroscopic classical measuring apparatus. To avoid bias, we assume that the device can choose any instant in the interval $\Delta t_m$ with equal probability.

---

Thus the source of indeterminism built into quantum mechanics is interpreted here as occurring due to the classical measuring device’s inability to measure at a precisely predefined time. We do not explain how the collapse of the wave function occurs when a measurement is made, only why the measurement outcome is probabilistic. Once a measurement is made, the wave function assumes the collapsed state.

Entanglement

Entangled states binding two or more qubits appear in our interpretation as the synchronization of the sub-Planck level oscillation of the participating qubits, as shown in Figure 3 for the two-qubit system Bell states, $((|00\rangle + |11\rangle)/\sqrt{2}$ and $(|01\rangle + |10\rangle)/\sqrt{2}$. A measurement on one of the entangled qubits will collapse both simultaneously to the respective state they are in at the instant of measurement (such as $\tau_1$ or $\tau_2$ in Figure 3). We do not know how Nature might accomplish the required synchronization.

It is, of course, clear that our interpretation cannot violate Heisenberg’s uncertainty principle since the latest measurement on a system collapses the system according to the measurement postulate. Thus, there can be no direct correlation between any earlier results of measurement on the system, and the succeeding measurement. Unlike the Copenhagen interpretation, in our interpretation it is not meaningless to ask about the state of the system in the absence of a measuring system.

10 Quantum Supremacy

So far we have benefitted from a euphoric rise in classical computing speeds tracing Moore’s law in historical time using the von Neumann architecture and CMOS hardware. While Richard Feynman had enthusiastically talked about “There’s plenty of room at the bottom”67 well ahead of his time at the annual American Physical Society meeting at Caltech on December 29, 1959 it did not arouse much attention at the time, but was well noticed in the 1990s. Quantum computing is both beyond Von Neumann architecture and beyond CMOS hardware. It represents an out of the world of classical physics transformative technology in computing. At the moment, building a universal, fault tolerant quantum computer still appears to be a distant goal.68

In the last few years, a flurry of activity has taken place in approaching quantum supremacy. Google and IBM top the list of competing contenders. In November 2017, IBM announced its 50-qubit processor, an improvement over its earlier 20, 15, and 5 qubit ones, respectively, in 2017, 2017, 2016. They used superconducting qubits operating at cryogenic temperatures.69 “IBM has been exploring superconducting qubits since the mid-2000s, increasing coherence times and decreasing errors to enable multi-qubit devices in the early 2010s. Continued refinements and advances at every level of the system from the qubits to the compiler allowed us to put the first quantum computer in the cloud in 2016.”70 Every processor has fault tolerance considerations accounted for. By the end of 2023, IBM hopes to have a 1,000–plus qubit device, called IBM Quantum Condor in operation.

17Feynman (1960).
18Villalonga (2020).
20Gambetta (2020).
Beyond that million-plus qubit processors. It is a part of an ambitious dream, “The future’s quantum computer will pick up the slack where classical computers falter, … to run revolutionary applications across industries, generating world-changing materials or transforming the way we do business.” Building such computers provide some of the biggest challenges in the history of technological progress. Such quantum computers will provide an unimaginable quantum leap to human progress.

In October 2019, Google made a “quantum supremacy” claim in a paper published in Nature, in which they claimed:

- Here we report the use of a processor with programmable superconducting qubits to create quantum states on 53 qubits, corresponding to a computational state-space of dimension \(2^{53}\) (about \(10^{16}\)). ... Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. ... A key systems engineering advance of this device is achieving high-fidelity single- and two-qubit operations, not just in isolation but also while performing a realistic computation with simultaneous gate operations on many qubits. [Internal citations omitted.]

Google’s claim was immediately contested by IBM, in a blog post, which claimed that an ideal simulation of the same task can be performed on a classical system in 2.5 days and with far greater fidelity, and not the 10,000 years as claimed by Google. They do have a point. As they point out:

The concept of “quantum supremacy” showcases the resources unique to quantum computers, such as direct access to entanglement and superposition. However, classical computers have resources of their own such as a hierarchy of memories and high-precision computations in hardware, various software assets, and a vast knowledge base of algorithms, and it is important to leverage all such capabilities when comparing quantum to classical.

The winner of the “quantum supremacy” contest in a combative sense is yet to be decided! There are technological hurdles to be overcome and mysteries to be resolved. A new experiment, known as the TEQ collaboration might throw some new light on our ignorance. “The overall objective of TEQ is the identification of the fundamental limitations to the applicability of quantum mechanics towards the establishment of a novel paradigm for quantum-enhanced technology that makes use of large-scale devices.”

References


Bera & Menon (2009). Bera, R.K., and Menon, V. A New Interpretation of Superposition, Entanglement, and Mea-

---

21 Gambetta (2020).
22 The term “quantum supremacy,” was proposed by John Preskill in 2012, and was meant to describe the point where quantum computers can do things that classical computers cannot. See Pednault, et al (2019).
26 Skibba (2020).
27 Project website http://tequantum.eu/


Dr. Rajendra K. Bera is the Chief Mentor at Acadinnet and a former honorary professor at the International Institute of Information Technology, Bangalore where he taught two courses: quantum computing, and intellectual property rights. In November 2005 he retired as the head of the R&D group of IBM Software Labs, India. He was with IBM during 1995-2005. Earlier he was a research scientist with the National Aerospace Laboratories, Bangalore (1971- 1995); a visiting assistant professor of aerospace, mechanical, and nuclear engineering, University of Oklahoma, USA (1979-1980); and a visiting faculty of aerospace engineering at the Indian Institute of Technology, Kanpur (1988) where he taught fighter aircraft design.

Dr. Bera received his BTECH, MTECH, and PhD degrees in aeronautical engineering from the Indian Institute of Technology, Kanpur, India. He was a member of the New York Academy of Sciences, and is a Fellow of The Institution of Engineers (India). He is listed in several Marquis Who’s Who.

He is the sole author of over 40 (out of 50) research publications in refereed journals and of several articles in the media. He is also the sole inventor on twenty eight (28) US patents as on 20 June 2012 and on a few more patent applications which are pending. His patenting areas include compiler optimization, resource allocation, pattern recognition, and static analysis of computer codes. His current research interests include pattern recognition in molecular biology, quantum computing, intellectual property rights, and non-linear dynamical systems.
GAMIFIED APPROACH TO LEARN ALGORITHMS

Exploring Basics of Algorithmic Approach

RAM P. RUSTAGI
Department of CSE, KSIT Bengaluru

VIRAJ KUMAR
Guest Lecturer, Department of CSA, IISc Bengaluru,

https://doi.org/10.34048/2021.2.F2
Introduction

The two-course sequence in Data Structures and Algorithms (DSA) in a typical Computer Science undergraduate programme assumes that students have some maturity in mathematical and computational thinking. At most institutions outside the top-tier, however, the majority of students struggle with basic skills such as manipulating inequalities and expressing algorithms that they know (e.g., determining whether a given integer N is prime or not) in some executable form (code or even pseudocode). As an example, when the first author asked second-year (and some third-year) students in his algorithms class to provide an algorithm for determining whether N is prime or not, the majority of the class proposed: “Check if N is divisible by 2, 3, 5 or 7. If yes, N is not prime, else it is prime.” We express this heuristic in pseudocode as follows (here \( \% \) is the modulus operator):

Heuristic(N)

\[
\text{if (N \% 2 == 0) or (N \% 3 == 0) or (N \% 5 == 0) or (N \% 7 == 0)}
\]

\[
\quad \text{return Not_Prime}
\]

\[
\text{else}
\]

\[
\quad \text{return Prime}
\]

It is possible that this heuristic was adequate for some students at the school level because they were only asked to test small values of N for primality. Indeed, when students were presented \( N = 143 \) or \( N = 221 \), several initially stated that these numbers were prime. When informed that \( 143 = 11 \times 13 \) and \( 221 = 13 \times 17 \), their immediate approach to fixing the above algorithm was to add additional checks to the if-condition: ... or \( (N \% 11 == 0) \), etc. Based on our collective teaching experience at multiple institutions outside the top-tier, we have no reason to believe that this group of students was unusually poor in their mathematical and computational thinking abilities. The recommendations of the National Education Policy (NEP 2020)[1] may, in time, provide school students with greater opportunities to hone these abilities. For the foreseeable future, however, we believe it is reasonable to expect many students will lack these abilities. Our Gamified Approach to Learning Algorithms (GALA) seeks to engage such students in the DSA courses by helping them overcome these deficiencies. This article builds on our previous article [2] and further demonstrates the GALA approach in two ways. First, we use games and puzzles to concretely demonstrate abstract mathematical insights that can lead to more efficient algorithms. Second, we use the GALA approach to motivate the study of problems that might initially seem uninteresting, but are valuable from a pedagogical perspective. We illustrate our ideas using examples from primality testing and properties of Fibonacci numbers, since most computing curricula expose students to these concepts.

1 Puzzle 1: Rectangular Arrangements

A classical interpretation of factorizing an integer N as \( p \times q \) is rearranging a single row of N blocks as a rectangle with p rows, each containing q blocks. For example, factorizing 33 as \( 3 \times 11 \) is shown in Table 1. A prime number is an integer \( N \geq 2 \) for which only the two trivial rectangular arrangements (1 row of N blocks, and N rows of 1 block) are possible.

Table 1: A rectangular arrangement of 33 blocks as 3 \( \times \) 11 blocks
To help students develop an algorithmic solution for testing primality (i.e., beyond the heuristic of testing divisibility by a small set of integers), we present the rectangular arrangements task as a puzzle, starting with \(N = 33\) blocks. We observe that most students are able to articulate one of the two correct solutions (3 \(\times\) 11 or 11 \(\times\) 3) quite easily. Next, we present \(N = 35\) blocks and ask students to “think of a solution in small steps”, starting with trying a rectangular arrangement with 2 rows. Most students recognize that such an arrangement will leave one block unused because \(N\) is odd. Thereafter, students typically try an arrangement with 3 rows (which fails) before successfully finding an arrangement with 5 rows (Table 2). (A few students also try an arrangement with 4 rows, failing to recognize that this is unnecessary.)

Table 2: An unsuccessful 3-row arrangement (left) and a successful 5-row arrangement (right) with \(N = 35\) blocks

Finally, we present \(N = 37\) blocks. Most students try out various unsuccessful rectangular arrangements (Table 3) before realizing that no arrangement is possible. In the subsequent discussion, we articulate two key points. First, we assert that no non-trivial rectangular arrangements exist precisely when \(N\) is prime. (Note that our three tasks have prepared students to accept this fact via inductive generalization – we believe that providing a deductive proof at this stage can be detrimental to understanding the overall algorithm for many of our students.) Second, we point out similarities between certain pairs of attempts. For instance, we represent the top-right arrangement in Table 3 (which has 5 rows) as \(37 = 5 \times 7 + 2\) and the bottom-left arrangement (with 7 rows) as \(37 = 7 \times 5 + 2\).

Table 3: Unsuccessful rectangular arrangements with \(N = 37\) blocks

This typically triggers a discussion where students recognize a symmetry that can be exploited for improved efficiency: if it is impossible to create a \(p\)-row arrangement for \(N\) blocks, it is also impossible to create an \([N/p]\)-row arrangement – hence it is unnecessary to check the latter arrangement. (Here, \([x]\) denotes the largest integer less than or equal to \(x\).) Once again, we rely on students to grasp this symmetry via inductive generalization from concrete values of \(N\) and \(p\). To avoid the unnecessary check, we point out that our algorithm can check \(p\)-row arrangements for progressively larger values of \(p\), until \(p\) becomes larger than \([N/p]\). At
this point, students are better prepared to understand the following naïve primality testing algorithm and the faster algorithm shown alongside (Table 4). Note that the while-loop executes \( N - 2 \) times for the former algorithm and less than \( (\sqrt{N}/2) \) times for the latter. We do not delve into a formal analysis of running time at this stage.

Table 4: Two algorithms for testing primality based on insights from Puzzle 1

<table>
<thead>
<tr>
<th>NaivePrimalityTest(N)</th>
<th>FastPrimalityTest(N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>if ( N \leq 1 )</td>
<td>if ( (N \leq 1) ) or ((N \geq 2 ) and ((N % 2 == 0)))</td>
</tr>
<tr>
<td>return Not_Prime</td>
<td>return Not_Prime</td>
</tr>
<tr>
<td>( p = 2 )</td>
<td>( p = 3 )</td>
</tr>
<tr>
<td>while ( p &lt; N )</td>
<td>while ( p &lt;= N/p )</td>
</tr>
<tr>
<td>if ( N % p == 0 )</td>
<td>if ( N % p == 0 )</td>
</tr>
<tr>
<td>return Not_Prime</td>
<td>return Not_Prime</td>
</tr>
<tr>
<td>( p += 1 )</td>
<td>( p += 2 )</td>
</tr>
<tr>
<td>return Prime</td>
<td>return Prime</td>
</tr>
</tbody>
</table>

Game 1: Coin-flipping

We now discuss a second example of using the GALA approach to concretely demonstrate abstract mathematical insights that lead to more efficient algorithms. Consider the following game between two players \( P_1 \) and \( P_2 \). There are \( N \) one-rupee coins [4] numbered 1 to \( N \) in a row with their heads down (tails up), as shown in first row of Table 5. The game with \( N \) coins involves \( N \) rounds, with each player (starting with \( P_1 \)) taking turns to complete an entire round. In the \( k \)th round, the player whose turn it is flips every \( k \)th coin (i.e., \( k \), \( 2k \), \( 3k \), ...). Thus, in the first round, \( P_1 \) flips every coin after which all coins now have heads up. In round 2, \( P_1 \) flips coin 2, coin 4, coin 6, ... (i.e., the even numbered coins now have heads down). The first 5 rounds of the game are shown in Table 5. The winner of the game is determined by counting the number of coins with their heads up after \( N \) rounds. \( P_1 \) wins if this number is odd. Otherwise, \( P_2 \) wins. After introducing students to the game, we ask them to determine whether they would rather be \( P_1 \) or \( P_2 \) for a given value \( N \). A naïve algorithm is, of course, to simulate the game for different values of \( N \). (Once \( N \) is fixed, the outcome of the game is deterministic.) We encourage students to implement the simulation as coding practice, but during the lecture hours we have students play the game manually in pairs with multiple values of \( N \) to discover a pattern. As with Puzzle 1, we claim no originality for this game (e.g., it appears as Locked Doors in Question 11 Exercise 1.1 [3]). Our contribution here is to develop the GALA pedagogy around such games and puzzles.

Table 5: The first 5 rounds of the Coin-flipping game
After about 15 minutes of play, we poll the class to see if they have noticed any interesting patterns during the simulations. Multiple pairs of students usually report that at the end of N rounds, the coins with heads up are numbered 1, 4, 9, 16, etc. Some students may be able to offer an explanation for why this happens (coin number k is flipped once for each factor of k, all integers k except perfect squares have an even number of factors, and an even number of flips leaves a coin in its initial configuration – heads down). However, we can once again proceed on the basis of inductive generalization that at the end of N rounds, the m coins numbered $1^2, 2^2, ..., m^2$ will have heads up, where $m = \lfloor \sqrt{N} \rfloor$. Thus, we are able to create a straightforward algorithm to determine the winner of the Coin-flipping game with N coins:

```python
CoinFlipWinner(N)
m = floor(sqrt(N))
if m % 2 == 0
    return P_2
else
    return P_1
```

The staggering simplicity of this algorithm compared to the cumbersome N-round simulation helps even the mathematically weakest of our students recognize that understanding the structure of the solution can lead to a much better algorithm. Thus, our approach has the potential to motivate such students to try and improve their mathematics fundamentals. Further, this algorithm can be revisited later in the course when discussing computational complexity, particularly to correct a common student misconception that highly optimized library functions (such as `sqrt()`) run in constant time.

2 Puzzle 2: Gadget Testing

Our purpose with this puzzle, which is adapted from Question 3 exercise 2.2 in [3], is to introduce students to the idea of “worst case” while analysing algorithms. We want to determine the highest floor $H$ of an N-storey building from which a certain type of gadget can be dropped without breaking (assuming $H \leq N$). We determine this highest floor experimentally, by dropping such gadgets from various floors according to some strategy. We are allowed to break a maximum of two such gadgets during these experiments, and we assume that as long as a gadget does not break, its structural integrity is unaffected and the same gadget can be reused for another experiment. Students are asked to come up with a strategy that minimizes the number of drops needed.

To help students better understand the problem, we spend some time discussing the “cautious” strategy: drop a single gadget from floor 1, 2, ... until it breaks or until we have dropped it from the top floor. Note that the gadget survives the first $H$ drops and if $H < N$, it will break on the next drop (by definition of $H$). Thus, this strategy requires $H + 1$ drops if $H < N$, and it requires $N$ drops if $H = N$. Hence, the strategy uses $N$ drops in the worst case. Notice that this strategy does not utilize the second device, so we encourage students to think of a more “aggressive” strategy that requires fewer than $N$ drops.

A natural variation, which students often suggest, is to drop a gadget from even-numbered floors: 2, 4, ... etc. until it breaks at some floor $2B$.¹ We already know that the gadget does not break when dropped from floor $2B - 2$, but we must do an extra drop from the “skipped” floor $2B - 1$. If the second gadget breaks when dropped from floor $2B - 1$, then $H = 2B - 2$. Otherwise, $H = 2B - 1$. This reduces the number of drops to $B + 1$. Many students are unable to express this quantity precisely in terms of N, but they do recognize that $B + 1 \approx N / 2$ in the worst case. At this point, at least one student suggests a further improvement to the strategy: drop the first gadget from every third floor: 3, 6, ... etc. until in breaks at some floor $3B$. Here, again, the second gadget is used to test the “skipped” floors (there are two such floors here). This strategy requires about $N / 3$ drops in the worst case. Students now immediately recognize that a bigger jump will produce an even better strategy. In their eagerness, however, students may overlook the fact that the “skipped” floors must be examined with the “cautious” strategy since one gadget is now broken. Thus, for a jump of size $k$, about $N / (k + k)$ drops are needed, where the latter term accounts for drops during the “cautious” phase.

Thus, the best strategy is the one for which the quantity $N / (k + k)$ is minimized by varying $k$. Here again, we find that only a few students have sufficient mathematici-

¹For a full analysis, we must also consider the case when the gadget does not break when dropped from any even-numbered floor $2B$, and examine two sub-cases: N is even and N is odd. We ignore this possibility here.
cal maturity to utilize their knowledge of calculus to determine that the minimum occurs for \( k = \sqrt{N} \). Hence, the optimal strategy requires about \( 2 \sqrt{N} \) drops in the worst case.

3  Fibonacci Sequence

Similar to prime numbers, numbers in the Fibonacci sequence have several interesting properties [5][6]. The first two terms in this sequence are defined as \( F_0 = 0 \) and \( F_1 = 1 \). Each subsequent term \( F_n \) is defined as the sum of the preceding two terms in the sequence: \( F_n = F_{n-1} + F_{n-2} \). Thus, the first few numbers in the Fibonacci sequence are: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... While there are several unexpected ways in which Fibonacci numbers show up in nature [7][8][9][10], we believe that the most compelling example arises from a property of these numbers in the following puzzle.

4  Puzzle 3: Missing/New Squares Paradox

Consider the square ABCD with sides of length 8 units shown in Figure 1. The area of the square is clearly \( 8 \times 8 = 64 \) square units. For this puzzle, we provide students with these instructions: First, cut the square into 4 parts: the two trapeziums APSR (with sides as AP = 5 units, PS = 5 units, and AR = 3 units) and DQSR (with sides DR = 5 units, DQ = 5 units, and DS = 3 units), and two right angle triangles PBC (with sides PB = 3 units and BC = 8 units) and CQP (with sides CQ = 3 units and QP = 8 units). Next, rearrange these four shapes as shown in Figure 2 to form a rectangle with sides of length 13 units and 5 units. Students are now confronted with a paradox: the area of the rectangle is 65 square units. Where did the additional square unit come from?

To deepen the mystery, we present students with another example (Figure 3) where the reverse happens: the four pieces of a square totaling an area of \( 13 \times 13 = 169 \) square units is rearranged to form a rectangle of only \( 21 \times 8 = 168 \) square units (Figure 4). This time, one square unit has vanished! Finally, we inform students that there is an infinite number of such paradoxes.

The mystery of missing and new squares can be understood as follows. To begin with, let us analyze the arrangement of pieces in Figure 2. The diagonal PS of rectangle consists of two parts PC and RS. These two diagonals have slightly different slopes (\( \tan^{-1} \frac{3}{5} = 36.9^\circ \) for PC and \( \tan^{-1} \frac{3}{8} = 21.8^\circ \) for RS), although to the naked eye they appear to align quite well. Thus, there exists a slight gap between the hypotenuse of the upper and lower triangles. Students are likely to attribute this gap to a slight inaccuracy in cutting the squares, but this narrow gap
has an area of precisely 1 square unit and accounts for the first paradox. A similar discrepancy in the slope of the hypotenuse of the two triangles \((\tan^{-1} \frac{3}{8} = 20.6^\circ)\) versus \((\tan^{-1} \frac{5}{13} = 21.03^\circ)\) causes the upper and lower triangles to overlap slightly in Figure 4. Once again, this overlap is easy to overlook, but it corresponds to the disappearing 1 square unit.

We find that these exercises spur students’ curiosity to find other such apparent paradoxes, and we bring up the topic of Fibonacci numbers and Cassini’s theorem [3] at this point:

\[
F_{n-1} \times F_{n+1} - (F_n)^2 = (-1)^n
\]  

(1)

Based on this theorem, an apparent paradox can be created with any three consecutive Fibonacci numbers \(F_{n-1}, F_n,\) and \(F_{n+1}\). Specifically, a square whose sides have length \(F_n\) can be cut into two trapeziums and two right angled triangles, and these pieces can be rearranged to form a “near rectangle” whose sides have length \(F_{n-1}\) and \(F_{n+1}\). The right-hand side in Cassini’s theorem indicates whether the “rectangle” has a new square (when \(n\) is even) or a missing square (when \(n\) is odd). We do not expect students to prove Cassini’s theorem, but we ask them to write an algorithm to verify that it holds for all \(n\) in the range 1 to \(N\) for a given integer \(N\). Our purpose in doing so is to convey a key point about reductions (discussed below), and we use the GALA approach to motivate the verification of Cassini’s theorem. Here is a naïve algorithm, which calls a helper function \(\text{Fib}(n)\) to compute \(F_n\):

```
VerifyCassini(N)

n = 1
while n <= N
  if n % 2 == 0
    if \(\text{Fib}(n+1) \times \text{Fib}(n-1) - \text{Fib}(n) \times \text{Fib}(n) != 1\)
      return Not_Checked
    else
      if \(\text{Fib}(n+1) \times \text{Fib}(n-1) - \text{Fib}(n) \times \text{Fib}(n) != -1\)
        return Not_Checked
  else
    return Not_Checked
n += 1
return Verified
```

Although this algorithm is extremely inefficient, it exemplifies a key concept that we expect students to master during their DSA courses: reduce the unknown problem (here, VerifyCassini) to a known problem (here, \(\text{Fib}(n)\), which we assume has already been discussed, either in the introductory programming course, or earlier in the DSA sequence). Since modern programming languages come with libraries of highly optimized implementations of data structures and algorithms, students must possess the ability to solve new problems by utilizing built-in data structures, and reducing the given problem to one for which an efficient algorithm already exists. Our central purpose in discussing the VerifyCassini problem is to highlight that a reduction may not always be the most efficient solution to the original problem.

In the above algorithm, most students are able to recognize that the expression \(\text{Fib}(n) \times \text{Fib}(n)\) wastefully calls the \(\text{Fib}(n)\) function twice with the same input. Hence, a more efficient solution calls the function \(\text{Fib}(n)\) once, saves the value in a variable (e.g., \(F_{\text{prev}} = \text{Fib}(n)\)) and then calculates the expression \(F_{\text{prev}} \times \text{Fib}(n)\). Developing this idea further, a few students typically point out that the values of \(\text{Fib}(n)\) and \(\text{Fib}(n-1)\) are computed while calculating \(\text{Fib}(n+1)\). Thus, a much more efficient algorithm that eliminates all calls to the \(\text{Fib}(n)\) function is as follows:

```
FastVerifyCassini(N)

n = 1
F_prev = 0
F_n = 1
while n <= N
  F_next = F_n + F_prev
  if n % 2 == 0
    if \(F_{\text{next}} \times F_{\text{prev}} - F_n \times F_n != 1\)
      return Not_Checked
    else
      if \(F_{\text{next}} \times F_{\text{prev}} - F_n \times F_n != -1\)
        return Not_Checked
  else
    return Not_Checked
n += 1
return Verified
```

FastVerifyCassini(N)
n = 1
F._prev = F._n
F._n = F._next
return Verified

Before closing this section, we note that a similar comparison of solutions with and without reductions can be considered with prime numbers. Assuming an efficient IsPrime(N) helper function, students can work on devising algorithms for the following problems [11]:

1. List all prime numbers less than a given number N.
2. List all twin prime numbers between two numbers M and N. Twin prime numbers are primes that differ by 2 (11 and 13, 17 and 19, 29 and 31, etc.)
3. Check if a number is circular prime i.e., whether all rotations of a number are also prime numbers. For example, 1193 is circular prime since all of its rotations 1931, 9311, and 3119 are also prime numbers.
4. Check if two consecutive Fibonacci numbers F_n and F_{n+1} for n>2 (since F_2=1 is not a prime) are relatively prime. Two numbers are said to be relatively prime if they have no common factor among them except 1.
5. Some of the Fibonacci numbers follow the property that number F_k is divided by k. For example, for k=1,5,12, F_1=1 divided by 1, F_5=5 is divided by 5, F_{12}=144 is divisible by 12. List first 10 Fibonacci numbers such that F_k is divisible by k.

5 Summary

We have proposed a gamified approach to learning algorithms – GALA. This article has demonstrated how our approach can help many of our students develop better mathematical and computational thinking abilities. In the next article, we will discuss recurrence relations and algorithmic efficiency in terms of time and space complexity.

References

10. Math is Fun (Fibonacci Sequence) https://www.mathsisfun.com/numbers/fibonacci-sequence.html
11. Programs for understanding properties of Fibonacci sequences. https://github.com/rprus tagi/GALA-Fibonacci-Number-Programs.git
Prof. Ram P. Rustagi

Dr. Ram P. Rustagi is currently working as Professor, CSE Dept, KSIT Bangalore, and honed up his academic skills with Ph.D from IIT Delhi, and M.Tech from IISc Bangalore. Prior to KSIT, at Cavisson Systems, he mentored new technology development using Machine Learning techniques in Security and Performance Monitoring. At PES University, he had taught Undergraduates, Post Graduates students, and successfully guided 3 Ph.D scholars. At PESU, he brought innovations in teaching computer network and security courses, and introduced practical experiential learning exercises.

Prof. Viraj Kumar

Dr. Viraj Kumar is a Visiting Professor at the Divecha Centre for Climate Change, IISc Bangalore and the Vice-Chair of ACM India's Special Interest Group in Computer Science Education (iSIGCSE). He was a consultant to the Committee to draft the National Education Policy (2017-18), and contributed to two education-related task groups of the Karnataka Knowledge Commission (2014-16). He holds a PhD in Computer Science from the University of Illinois at Urbana-Champaign.
A LAT-LONG CODE

SRINIVASAN RAMANI

https://doi.org/10.34048/2021.2.F3
Geographical Location information is of increasing value. Almost everyone carries a cell phone. A lot of them are equipped with GPS. Map servers provide good maps and driving instructions. However, there is no widely used coding system for locations easy to use by humans as well as machines. So, now you need to spend several minutes to communicate your destination properly when you are calling for an app-based taxi service, or placing an order online for something you wish to buy. Ideally, there should be a machine readable address code in your cell-phone contacts and you ought to transfer it to an app merely by clicking the contact.

1 Desired Properties of any Location Code

Ideally, any system for use by humans should have the following properties. Knowing the current address and destination, it should be easy to estimate the distance manually. I should also know in which rough direction I should go. The address should be short and should be usable irrespective of the language of the user. I should be able to read it over a phone call. An app on a cell phone should find its own location and that of any desired destination in this code. The coordinate system must be adequately accurate. When I get to the destination, I should be close enough to where I want to go. I should not have to ask for help from strangers.

The system should in principle be accurate enough for driverless cars as well.

We are almost ready to implement a suitable address-coding system. What we need now is a design that is easy to learn and to use. User friendliness is the key. Look at bing/maps or Google maps; you can type in something like “Bengaluru Lalbagh” and locate it on the map. Right click on it and you will get its latitude and longitude coordinates. I list three locations with their coordinates to show the format as it displays coordinates all over the world. The numbers are rounded off to four places after the decimal.

Ashoka Pillar, Bengaluru 12.9439, 77.5852
Sydney Opera House -33.8566, 151.2152
Honolulu, Kahanamoku Beach 21.2836, -157.8392

Consider a system that uses 8 bytes for Latitude and 9 bytes for Longitude. What about locating the address given a truncated lat-long code as shown above? Both map systems mentioned above mapped these codes into a physical location. One did it accurately and the other showed some inaccuracy. I am sure that the inaccuracy can be fixed easily.

A cell phone number with a two-digit country code is 12 numerical digits long, for comparison. So, the lat-long code is not very long. The proposed system can use existing GPS hardware and software on smartphones effectively. It is a worldwide coordinate system. So, users can use a single app worldwide without any changes. The code’s novelty is in being easy to implement and to use.

Consider a location P that has latitude and longitude as follows, using a total of 16 numerical digits plus one comma symbol:

Latitude (8 digits) = Sxx.xxxx and
Longitude (9 digits) = Syyy.yyyy

S is the + or - sign. The proposal is to write northern latitudes as +13, +45, etc. and to write southern latitudes as -13, -45, etc.

We propose that the LatLong address of a given point P be written as

Sxx.xxxx,Syyy.yyyy

2 Features of this Naming Convention

1. A suitable app on a smart phone can find the Lat-Long code of any given place on a map. It can point out, on a map display, a location referred by given coordinates.

2. Given two points described by Lat-Long codes, it is easy to perceive the direction of one point from the other. It is also easy to estimate the approximate distance of one from the other manually. An app can compute and display effective travel directions.

3. The resolution of this system varies at different latitudes, but is better than 15 meters everywhere. The latitude and longitude circles defined by any point’s latitude and longitude are at most about 40,000 KM in length. If a location is represented on one of these circles using the notation described, the resolution offered in the east-west and north-south directions would each be approximately 10 meters.

4. It is common in many countries to limit the resolution offered by navigational aids to civilians. The coding system described here needs only map data in the public domain. So, it conforms to usual security discipline.

5. The student project I suggest is to create an app. It will help locate places on a map, get and use ones
own cell phone location. It will also give driving directions. Google allows developers to write apps interfacing with their code and adding value. We could do that to simplify app building.

6. It is worth noting that in latitudes like those in India, 1 degree of latitude or longitude is about 110 km. When the latitude and longitude are known in the format described above, maps can show it with an error less than 15 meters.

Srinivasan Ramani

Dr. Ramani was Research Director, HP Labs India, located in Bangalore. He founded the National Centre for Software Technology (NCST) in 1985, and had directed it during 1985-2000. His work at NCST covered R & D in the areas of computer networks and knowledge based computer systems. He made significant contributions to the creation and development of the Indian academic network, ERNET, which brought Internet connectivity to India in 1988, and the Bombay Library Network, Bonet. He has served as Editor, Journal of Information Technology for Development, for a number of years.

He was President, International Council for Computer Communication, and Chairman of the Governing Board of the Information Library Network (INFLIBNET). He was also a member of the High Level Panel of Advisors to the UN on Information and Communication Technologies.
MEDIA KIT

Make accessible to engineering students, faculty and engineering practitioners, the advanced computing and communication topics

READER PROFILE

Engineering college students and Professors
Masters and Doctoral students
Engineering practitioners

<table>
<thead>
<tr>
<th>SL NO</th>
<th>Page</th>
<th>Cost in ₹</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Full Page</td>
<td>70,000</td>
</tr>
<tr>
<td>2</td>
<td>Half Page</td>
<td>40,000</td>
</tr>
<tr>
<td>3</td>
<td>Double Spread</td>
<td>140,000</td>
</tr>
<tr>
<td>4</td>
<td>Inside Front Cover</td>
<td>125,000</td>
</tr>
<tr>
<td>5</td>
<td>Inside Back Cover</td>
<td>100,000</td>
</tr>
<tr>
<td>6</td>
<td>Back Cover</td>
<td>1,50,000</td>
</tr>
</tbody>
</table>

Note: Taxes are applicable

READERSHIP
PRINT: 20,000 - 25,000
ONLINE: 75,000-100,000

SUBSCRIPTION

Category A: Institutional for 5 years ₹5,000/- plus 3 copies of ACC
Category B: Non-institutional for 5 years ₹1,200/-
Category C: Non-institutional for 3 years ₹1,000/-

To place an advertisement please contact
+91 76249 57142    +91 80 2360 7142    acc@accsindia.org
COMMEMORATING
THE PLATINUM JUBILEE OF ENIAC

V. RAJARAMAN

Supercomputer Education and Research Centre

Indian Institute of Science, Bengaluru

https://doi.org/10.34048/2021.2.F4
1 Introduction

On February 15, 1946, ENIAC (Electronic Numerical Integrator and Computer) was formally unveiled at the Moore School of Electrical Engineering of the University of Pennsylvania at Philadelphia in the United States of America. This year marks its 75th anniversary. ENIAC was a landmark in the history of computing as it was the first electronic computer that worked reliably. It was used at first to tabulate artillery firing tables and later a refurbished model was used to solve several important problems including Monte Carlo simulation, weather prediction, and fluid dynamics problems. All earlier computers such as Complex Number computer built by George Stibitz at Bell Telephone Laboratories in 1940, Z3 by Konrad Zuse in 1941, and Harvard MarkI built by Howard Aiken in 1944 used electromechanical relays. Atanasoff and Berry built an electronic computer in 1942. It was, however, not general-purpose and was incomplete. In this article, I will recount the history of the design and development of ENIAC.

2 Genesis of ENIAC

The United States of America did not participate in the second world war when it began in 1939. It was, however, helping the United Kingdom and its allies with armaments and logistics. It entered the war when Japan bombed Pearl Harbor in 1941. One of the important wartime efforts of the US army was to calculate artillery firing tables using trajectories of shells and bombs. There were thousands of artillery guns and the task was arduous. Tables for artillery were prepared at the US army’s Ballistics Research Laboratory (BRL) at Aberdeen in Maryland. A mechanical differential analyser designed by Vannevar Bush of the Massachusetts Institute of Technology was used in preparing the tables. The Moore School of Electrical Engineering at the University of Pennsylvania at Philadelphia owned a similar, more powerful, differential analyser. The university was a few hours’ drive from Aberdeen. The army decided to contract out the preparation of some artillery firing tables to the university. It appointed Lieutenant Herman Goldstine, who had a doctorate in mathematics from the University of Chicago and had been inducted into the army, to liaise with Moore School and oversee the progress of the project. As there was considerable work, the university also employed around 200 ladies who were trained to use mechanical desk calculators to prepare the tables. Adele Goldstine, Herman’s wife was a qualified mathematician, having obtained a Master’s degree in mathematics from the University of Michigan, and assisted in selecting and training the ladies. These ladies were called “computers”. The calculation of the tables was laborious. It took 20 hours of manual calculation to tabulate a 60-second shell trajectory. The differential analyser was faster and took 15 minutes for the same task. As the number of tables required was huge the project was behind schedule. A faster device for calculation was required.

Moore School of Electrical Engineering at the University of Pennsylvania had a group of engineers who were teaching electronics. Among them was John Mauchly, who was interested in developing electronic computers. He had obtained a Ph.D. in physics from Johns Hopkins University in 1932. He taught physics at Ursinus College, a small liberal arts college in Pennsylvania. He was an extrovert and liked to learn about new developments. In June 1941, he had visited John Atanasoff who was building a digital computer with vacuum tubes at the University of Iowa to solve simultaneous algebraic equations. Atanasoff and his student Clifford Berry had used binary numbers and Boolean logic in designing their machine. The computer called Atanasoff-Berry Computer (ABC) was not fully functional as war intervened and the navy drafted Atanasoff. Mauchly saw how vacuum tubes were used to perform arithmetic operations and took copious notes about the design of the machine [1]. When he returned from Iowa, he was pleasantly surprised that he had been admitted to a wartime training course in electronics at the Moore School. The course was about the design of vacuum tube circuits and J. Presper Eckert was an instructor. After the course, Mauchly was offered a position of an instructor in electronics at Moore School. He accepted the offer as he was interested in designing electronic circuits. He became friendly with Eckert who he knew was an excellent designer of electronic circuits. Mauchly was familiar with the differential analyser and its use in calculating artillery tables. He was also aware of its slow progress. He discussed with Eckert his idea of designing an electronic computer to speed up the calculation. They teamed up and proposed to John Brainerd, the head of Moore School, that an electronic computer be built to calculate the artillery tables. Brainerd agreed. Eckert and Mauchly wrote a proposal in early 1943 requesting funding of $61,700 from the US army to build Electronic Numerical Integrator and Computer (ENIAC). Goldstine who was liaising with the group at Moore School supported the proposal. The US army agreed to fund the project and released funding in June 1943 and classified the project as top secret.

3 Design of ENIAC

Eckert and Mauchly started constructing ENIAC as soon as the money was released. Mauchly was the designer of the logic of the system. Eckert designed the electronic circuits and built the computer. The computer stored decimal digits in accumulators. Each accumulator was 10 digits long and there were 20 accumulators. Counters
were used for arithmetic operations and accumulators stored the results. ENIAC could add, subtract, multiply, divide, and find the square root of a number. It had three function tables to store values of arbitrary functions. The circuits were driven by a clock that gave a pulse once every 10 microseconds. It could add 5000 and multiply 333 ten-digit numbers in one second. Data were input to the computer using an IBM card reader and the results of the computation were punched on an IBM card punch. An offline printer printed the punched results. ENIAC did not have a memory. It was programmed by interconnecting its units by plugging wires on a large 1 square metre plugboard. It was tedious and required a good knowledge of the logical structure of the computer. It took weeks to set up a program to solve a problem and debugging was difficult. As it was primarily used in its early days to tabulate artillery tables this was not a serious disadvantage. It could be programmed to perform looping, conditional branches, and subroutine calls. Six mathematically savvy ladies were selected from among the ladies operating the desk calculators and trained to program the computer. Their contribution was invaluable in using ENIAC to solve problems. Unfortunately, at that time, their importance was not recognized and they were not even invited to the dinner that was given to celebrate the successful completion of ENIAC [2]. Belatedly, in 1997 they were inducted into the Women in Technology International Hall of Fame.

ENIAC was constructed one unit at a time. As the construction progressed, the project was frequently running out of funds. On the recommendation of Goldstine, the project was periodically renewed and funds released. When it was completed, it cost a little over $ 485,000 and used around 18,000 vacuum tubes, 7200 diodes, 1500 relays, 70,000 resistors, and 10,000 capacitors. It weighed 27 tons, occupied 1800 sq. ft. room, and consumed 150 kW power. It required a massive air circulation system to cool it.

One of the major problems encountered during its use was the failure of vacuum tubes. The average life of a vacuum tube was 3000 hours. With 18,000 tubes the probability of failure of one of the tubes and consequently the computer was around 10 minutes. Eckert increased the life of vacuum tubes a hundred-fold by operating them at two-thirds rated voltage. The filaments of the tubes fail if they are turned on and off frequently. Eckert decided to keep the computer always on [3]. With these changes, the computer stabilized and used to fail only once a day. Procedures were found to quickly locate the bad vacuum tube and replace it. ENIAC could tabulate the 60-second trajectory of an artillery shell in 30 seconds, 30 times faster than the differential analyser.

4 Von Neumann’s Involvement With ENIAC

John Von Neumann was a famous mathematician who was a professor at the Institute for Advanced Studies at Princeton. He was a consultant to the US army and was a frequent visitor to BRL at Aberdeen. He was interested in using computers to solve mathematical problems related to the design of the atomic bomb in which he was involved. In this connection, he had visited the computing laboratory at Harvard University and the Bell Telephone Laboratories. Both these computers were too slow to solve his problem. Goldstine met him by accident at the Aberdeen railway station in August 1944. He was in awe of the great mathematician and built up the courage to talk to him. He was pleasantly surprised when Von Neumann evinced a keen interest in computing. Goldstine briefly described ENIAC and mentioned that it used vacuum tubes and multiplied 333 ten-digit numbers per second. Von Neumann who had seen slow relay computers was interested in knowing about the fast electronic computer. A visit was arranged for him to see ENIAC. He went to Moore School later in August and was requested to be a consultant to the ENIAC group. In 1944, Eckert and Mauchly were exploring how to design a memory for ENIAC and use it to ease programming. When Von Neumann joined the group, the discussion was steered towards programming ENIAC. During several brainstorming sessions, the idea of storing program and data to be processed by the program in the main memory of a computer emerged. The construction of ENIAC had progressed far and it was too late to incorporate in it a memory to store programs. Even though
Eckert had an idea of designing a memory using mercury delay lines it would have taken too long to realise it. It was decided to incorporate this idea into a successor computer EDVAC (Electronic Discrete Variable Automatic Computer).

Extensive notes were taken during these discussions. Von Neumann used these notes and during a long train journey to Los Alamos wrote a document describing the logical structure of the proposed computer. It was titled “First draft of a Report on the EDVAC”. He sent it to Goldstine. Goldstine got it typed and attributed the document solely to Von Neumann, even though it was the result of a group effort. Several copies were made and distributed in June 1945. This document described in detail the architecture of a stored-program computer as one consisting of five units; an input unit to read program and data, the main memory in which the program and data are stored, a control unit that interpreted instructions of a program and ordered the other units to carry them out, an arithmetic unit to carry out arithmetic operations, and an output unit to print the results. This was one of the most important documents in the history of computing as it steered the development of all future computers to this day. The architecture came to be known as the “Von Neumann architecture” even though the idea was the result of brainstorming that took place with several participants including Eckert and Mauchly.

5 ENIAC – Second Avatar

A dispute arose between Eckert - Mauchly and the University of Pennsylvania on the patent rights of ENIAC. It led to Eckert and Mauchly leaving the university in March 1946. They started a company named Electronic Control Company, that was renamed Eckert–Mauchly Corporation, to manufacture computers. They also took along with them many engineers and programmers from the ENIAC team. Those who were left continued to use ENIAC.

In December 1946, the US army that had sponsored the design and construction of ENIAC decided to shift it from Moore School to BRL at Aberdeen. ENIAC weighing 27 tons and with 18,000 delicate vacuum tubes was disassembled, packed, and shipped to Aberdeen; an arduous task! A group of engineers at BRL reassembled it which was difficult and took long. It was quite a challenge to get it back to a working state. After seven months, in late July 1947 ENIAC was switched on and started working. Between December 1946 and July 1947, Von Neumann and Adele Goldstine started experimenting methods for simplifying programming ENIAC as wiring the plugboard for each program was inflexible, tedious, and took weeks. ENIAC had three function tables each storing 1200 digits. They decided to re-purpose these function tables and use them as a read-only memory in which programs can be stored. Pulses generated from these tables were to be used to control the sequencing of calculations. A set of around 80, two-digit instructions (called order code) was designed by the group that included Richard Clippinger of BRL. Von Neumann suggested how to re-purpose some accumulators to work as program counter and as a single address computer. The hardware of ENIAC was modified by adding a unit to facilitate the use of function tables as program memory. The modified ENIAC initially failed to execute programs. The engineers slowed down the computer by reducing the clock speed and it worked [4]. ENIAC started working almost like a stored program computer in March 1948. It was used to successfully run simulation programs (using what is known as Monte Carlo method) to aid in the development of the hydrogen bomb. With this success, ENIAC was used for solving numerous problems such as weather prediction and wind tunnel design. Programming a new problem took days rather than weeks. ENIAC was the fastest and most easily programmable computer in the world between 1948 and 1950.

ENIAC was the workhorse for the US defence department during the late 1940s. It was improved with additional hardware, including a high-speed shifter in 1952 and a 100-word random access magnetic core memory built by Burroughs Corporation in 1953. Meanwhile, faster and more versatile commercial computers were entering the market. ENIAC was shut down on October 2, 1955. It is now in the Smithsonian museum in Washington D.C. The US post office released a stamp with the picture of ENIAC in 1996 on its golden jubilee.

Acknowledgment

I thank Dr. N. Dayasindhu for reviewing this article and his critical comments that improved it.

References

Dr. Vaidyeswaran Rajaraman obtained his MTech from IIT Kanpur in 1965 and later his PhD. He pioneered computer science education in India with the introduction of BTECH program in IIT Kanpur. with Computer Science as an option, the first time the subject was offered as an academic discipline in India. He moved to Indian Institute of Science, Bangalore and developed low-cost parallel computers and a supercomputing facility of which he served as Chairman from 1982 to 1994. He has published over 70 scientific papers in national and international peer-reviewed journals and 23 text books. Dr Rajaraman has been awarded the Padma Bhushan and the Shanti Swarup Bhatnagar Prize for his contributions to the field of computer science.
YOU CAN ALSO ACCESS DIGITAL VERSION OF THIS JOURNAL ONLINE

https://journal.accsindia.org