February 2013, Vol 2

# HITech

***Technical magazine of CSE Department, Heritage Institute of Technology***

***Editor:*** *Reshma Roychoudhuri, Assistant Professor, CSE*

## In This Issue:

# Foreword

**by Reshma Roychoudhuri ,**
**Assistant Professor, CSE**

"If we consider all the people who have won Nobel Prize so far, the probability of a person winning the Nobel is 1 in $10^{7}$" Thus started the twenty minute speech of Prof Dr Ei-ichi Negishi, 2010 Nobel Prize winner in Chemistry, at the auditorium of the Heritage Institute of Technology on the 5th of January 2013. Prof Negishi was here as a guest for the Indian Science Congress (ISC) and Heritage being one of the venues for the same had arranged for two events; one being the lecture by Prof Negishi and the other a 'Children meet Scientist' initiative.

As a school boy young Negishi was more of an outdoor person, till the time when he injured his right hand and was forced to stay at home. With nothing much to do at home he decided to concentrate wholly on studies for a change. And voila, he was soon in the top 3 students of the school! As the school topper he got an opportunity to sit for an entrance examination to the University of Tokyo. Young Negishi cracked the exam and was accepted at the University of Tokyo. From the University of Tokyo he went on to win a scholarship to the University of Pennsylvania and that started his journey in America. Later, as a post doctoral student with Herbert C Brown, professor of chemistry at Purdue University, Dr Negishi got an invaluable one-to-one coaching which he thinks to be the deciding factor in his winning the Nobel. In his own words, if one's focus is continuous academic improvement then "After going through college, graduate school and perhaps getting a PhD degree, one should seek one-on-one coaching from one of the world's best individuals in the field of one's interest." As we, the faculty, students and staff, sat enchanted, Prof Negishi went on to recollect many such memories starting right from his pre-school days which, according to him, led to his successful Nobel endeavor.

In the last few months, since the first edition of HITech got published, the Department of Computer Science and Engineering saw quite a few notable events. In a workshop on 'Selected Research Topics in Computer Science' organized by the Department eminent academician Dr. Vishwani D. Agrawal, Professor, Department of Electrical and Computer Engineering, Auburn University spoke on 'Power and Time Tradeoff in VLSI Testing'. Also present were Dr. Sandeep Shukla, Professor, Bradley Department of Electrical and Computer Engineering, Virginia Tech who spoke on 'A Tutorial Introduction to Embedded Software Synthesis'; Dr. Partha Pratim Das Professor, Department of Computer Science and Engineering, IIT Kharagpur who spoke on 'Video Assisted Analysis of Infant Neurological Examinations' and our very own Dr. Prosenjit Gupta, Professor, Department of Computer Science and Engineering, Heritage Institute of Technology who spoke on 'Computational Geometry, Range Searching and Aggregation'. A Faculty Training by Industry under Industry-Institute Interaction program of TEQIP was also organized in which IT professionals from TCS gave presentations on Wireless Networking, Big Data and Cloud computing. Students from HIT participated in the Master in Mainframe contest 2012 organized by IBM. Md. Aman Khan of 3rd Year (CSE) made the Department proud by winning a certificate of appreciation from the organizers.

2013 started with a workshop on Complex and Social Networks which saw eminent speakers like Dr Niloy Ganguly[IIT KGP], Dr Animesh Mukherjee [IIT KGP], Dr Aparna Basu [Emeritus scientist, CSIR- NISTADS], Dr S. S Manna [SN Bose National Centre of Basic Science, Dr Amit A. Nanavati [IBM- India Research Lab] and Dr. Sitabhra Sinha [Institute of Mathematical Sciences] share the podium. HIT ACM Students Chapter organized it's first event of 2013, Brainstorm, a quiz based on logical reasoning which saw 146 registrations.

This edition has an interesting collection of articles with the focus more on interesting applications of Computer Science than on core technical topics, which I am sure will appeal to everybody. Also, would like to remind all CSE students and staff that

HITech depends on your contribution. So do send us your articles.

I would like to end this article with an anecdote from the 'Children meet Scientists' event held in HIT auditorium as part of the Indian Science Congress. Eminent scientists introduced themselves and were then asked questions from the school students in the audience. Interestingly the first question asked, by a boy not more than 7 years old and with thick glasses, was "Did you choose to become a Scientist or your parents chose it for you?" Even though the auditorium broke into laughter it was also a reality check for all of us. When was the last time we asked ourselves if we are working on a subject that we love to do? If we haven't then lets ask now!

Wish all readers a happy reading.

# Grisha Perelman – an Enigma

**by Dr. Subhashis Majumder, Professor and HOD, CSE**

In the year 1900, David Hilbert, a famous German mathematician, presented a collection of 23 unsolved problems that set the course for mathematical research of the 20th century. After a century of great effort, most of the problems have been solved except for a few very notoriously difficult ones like the Riemann Hypothesis. To commemorate the 100th anniversary of Hilbert's announcement of his problems, leading mathematicians in 2000 comprised a list of 7 open problems which are currently considered the most important for mathematics. These are known as the Millennium Problems, and each problem carries a prize of $1 Million for its resolution, awarded by the Clay Mathematics Institute. The Riemann Hypothesis also appears in this list, the only problem to appear in both lists. Among other famous problems, this list also includes the famous P vs NP problem from the domain of Computer Science. The only Millennium Problem that has been solved till now is the Poincaré Conjecture.

The person who resolved the Poincaré Conjecture in 2002 after it had remained open for nearly a century is popularly known as Grisha Perelman, but he does not like the idea of the media calling him Grisha, a customary shortening of the Russian name Grigori. Perelman was born in 1966 in the former Soviet Union to Jewish parents. His mother had given up her own graduate studies in order to raise him, a rare sacrifice in the western world. Perelman's mother enrolled him in an after school mathematics club, as in our childhood we went for drawing, singing or dancing and sometimes for recitation. The trend has changed towards cricket club and chess club and from the last decade towards public singing and dancing but still we as a society do not take mathematics seriously! We do not send our children to mathematics club, and do not believe that mathematics can be fun.

In the after-school mathematics club Perelman got his training under Sergei Rukshin, a mathematics undergrad from Leningrad University. Rushkin had a troubled youth but was passionate about mathematics and was only nineteen when he first met Perelman. Rukshin found that Perelman even as a kid was unusually precise in his mathematical thinking. As he was approaching the finishing days in his school, Perelman had to plan for his admission into some university and in those days for a mathematics major Leningrad University was the most prestigious option. However for a boy of Jewish descent, this option was quite a difficult one. Either he had to be one of the two Jews accepted at Leningrad University every year or he had to be one of the members of the Soviet team for the International Mathematical Olympiad, which guaranteed automatic admission to Leningrad University. Perelman opted for the latter choice. Alexander Abramov who ran the training program for the International Mathematics Olympiad, commented that Perelman never faced a problem in a competition that he could not solve. He won a gold medal in the 1982 International Mathematics Olympiad with a perfect score that ensured his entry into Leningrad University.

After finishing his undergraduate and graduate studies in Leningrad University, Perelman got a post-doctoral position at the Steklov Institute of Mathematics at Leningrad. After this, in the late eighties and the early nineties, Perelman worked in Riemannian Geometry in several US universities like NYU and Stony Brook and at UC Berkeley as a post-doctoral fellow. He gave an elegant proof for a theorem in Topology called the 'Soul Conjecture', that immediately made him a star and he was offered positions at both Stanford and Princeton University. He refused both the offers and especially declined the Princeton Maths. Department offer as they had asked for his CV. In the summer of 1995, he returned back to the Steklov Institute for a research only (i.e. no teaching) position.

The Poincaré conjecture, proposed by French mathematician Henri Poincaré in 1904, was one of the most famous open problems in mathematics. A 3-sphere is a 4-dimensional analogue of a normal

sphere. It consists of the set of points equidistant from a fixed center in 4-dimensional Euclidean space. Now any loop on a 3-sphere can be contracted to a point. Poincaré conjectured that for any closed three-dimensional manifold such that any loop can be contracted to a point is topologically same as a 3-sphere. The interesting thing is the analogous result has been known to be true in higher dimensions ( > 4) since 1960. The four-dimensional case was finally solved in 1982 by Michael Freedman. However, the case of three-manifolds turned out to be the hardest, probably because for topologically manipulating a three dimensional manifold, very few dimensions are available to move the "problematic regions" out of the way without interfering with something else. It needed the brilliance and dedication of Perelman to conquer this final frontier.

After 1995, Perelman almost went into oblivion and after a patient effort of 7 long years published three papers between November 2002 and July 2003. Unlike the usual practice in his field he did not send them to peer-reviewed journals but instead preferred to put them on an internet archive for verification. Then he started sending abstracts to some famous mathematicians with whom he had kept no contact for more than a decade. But slowly people understood the advanced techniques he used in his proof using Ricci Flow and it was found that it is correct. Such careful verification was necessary as there had been many failed attempts over the last century to tackle the Poincaré conjecture and many lives were spent in failed attempts.

But the fact that attracts the scientific community and also the society at large is not his astounding brilliance but his apathy towards accepting awards and recognition. In 1996, he turned down a prestigious prize from the European Mathematical Society saying that he did not think that the prize committee was qualified enough to assess his work, even positively. Then in 2006, Perelman was awarded the Fields Medal by the International Mathematical Union for his resolution of the Poincaré conjecture. The Fields Medal is the highest honor in the field of Mathematics where there is no Noble Prize. However he declined to accept it, even after Sir John Ball, President of the International Mathematical Union tried to persuade him for 10 long hours. He was the first and till now the only person to decline this prestigious award. He said that the prize was completely irrelevant for him. He also said, “Everybody understood that if the proof was correct then no other recognition is needed. ... I'm not interested in money or fame....'I don't want to be on display like an animal in a zoo. I'm not a hero of mathematics. I'm not even that successful ”. Finally on 18th March 2010, Perelman was awarded the Millennium Prize of 1 million dollars. He did not attend the ceremony in Paris in June 2010, held in his honor and in July 2010 he formally refused to accept the Millennium Prize. Among other details he stated that the main reason was his disagreement with the organized mathematical society.

Perelman stopped working at the Steklov Institute since 2003 and according to an interview given in 2006, he was unemployed then. He currently finds Mathematics to be a painful topic to discuss and many of his acquaintances speculate that he might have left Mathematics altogether. Since 2003 his interactions with fellow mathematicians have been on the decline and even with Rukshin it has come down to a telephone conversation once per year. Very recently in 2012, Brett Forrest had a chance to have a talk with him for a few minutes after waiting for three days near the gate of the apartment complex where Perelman stays. When asked what he is doing now, Perelman said, “I have left mathematics. And what I’m doing now, I won’t tell you.” Perelman now stays with his mother in a small apartment in St. Petersburg and lives out of his mother’s pension that amounts to about 160 USD per month. Once the residents of that apartment complex found that there had been a sudden escalation in the number of cockroaches in that complex. A little search effort led to the source, which was a mattress in Perelman’s apartment and in fact it was the one that Perelman used himself. Many things were said about Perelman’s mental set-up and it might be an interesting topic of analysis for the applied psychologists. However I only think how fortunate were those cockroaches, some of them might turn out to be mathematicians in their next life!

Masha Gessen, his biographer and also his contemporary, on the basis of many incidents of Perelman's pickiness, his long hair and fingernails and Rasputin-like appearance, and his often asocial behavior, suggests that he has Asperger's syndrome, sometimes referred to as autism-lite. Quoting the psychologist Simon Baron-Cohen, an expert in the field, she writes that people with Asperger's have limited social skills, have trouble communicating, often speak oddly, and frequently need help with the minutiae of everyday living and so are dependent on others, such as their mothers, as was the case with Perelman. I wonder sometimes how much our present society has dried up that a personality like Perelman could not find a second woman to depend on.

Moreover, these people are extraordinarily good at systematizing but extraordinarily poor at empathizing, and have what Baron-Cohen calls an "extreme male brain." But Brett Forrest observed that the case was not always like that. As a child, Perelman interacted with other students and also enjoyed ping-pong and the opera. However Rukshin also said, "If Grisha ever looked at anything with loving eyes, it was on the blackboard." Perelman was quoted in an article saying that he was disappointed with the ethical standards of the field of mathematics. The article in 'The New Yorker' implied that Perelman was referring particularly to the efforts of Fields medalist Shing-Tung Yao to downplay Perelman's role in the proof and play up the work of Cao and Zhu. But actually even before that he was turning inward. In 1995, after returning to Russia he told his childhood mentor, "It's possible to sell a theorem and it's possible to buy it. Even if you don't have anything to do with it." However, whatever is the case, Perelman seems to have a perpetual apathy towards branded positions; as recently in 2010 he refused to participate in the 'Russian Silicon Valley' project. When pointed out that one of the leaders of the project will be a Nobel Laureate in physics, Perelman was not interested to change his decision. What can be said about such a person who refuses to bow down in front of recognition, for which the whole society is running around so eagerly. Alburt Camus once commented about the central character of his own famous novel 'The Outsider' that the specialty of that character is he cannot compromise with truth. Even when confronted with the possibility of death, that character who was a murderer himself could not utter a lie taught by the priest. Perelman sometimes gives me the feeling that he is such an outsider who seeks only truth, whatever be its outcome. I feel sages are not only born in India, they can grow out of thoroughly materialistic societies like his. If I ever get a chance to meet Dr. Perelman, what can I tell him? I have no words of my own. Have to borrow from our great poet –

'Je dhone hoia dhoni, monire manona moni
Tahari khanik, magi ami noto sire'

Note: The contents of this text were mostly taken from some write-ups freely available over the internet, like Wikipedia, and some other small interviews given by Dr. Grigori Perelman.

# Randomness Redefined

**by Sourav Kr. Bose, 2nd Year, CSE**

What is a Random number? Is there some sequence of numbers, which are truly random?
At the first sight it may seem easy to define randomness, in practice it proves to be quite difficult.
A random number is a number generated by a process whose outcome is unpredictable and which cant be subsequently reproduced. But over time no such algorithm has been designed that can generate a truly random number. What most programming languages and mathematical tools provide us is some pseudo random number. So, in that case we may say, a random number is somewhat relative. Some mathematically defined sequences, such as the decimals of pi, exhibit some of the same characteristics as random sequences, but because they are generated by a describable mechanism, they are called *pseudorandom*. To an observer who does not know the mechanism, a pseudorandom sequence is unpredictable.

**HISTORY :**

Life –was a game of fate and fate was synonymous with chance, which, in turn, co-existed with randomness, such had been the face of history. The very popular game of dice originated from this concept of destiny.
Towards the late twentieth century, ideas of algorithmic information theory had become most lucrative of the 'then state of affairs' by way of the theory of algorithmic randomness…

**APPLICATION OF RANDOM NUMBERS:**

- Game Programming.
- Statistical Sampling.
- Simulation.
- Mathematics.
- Finance.
- Cryptography.

Game Programming: Games like Ludo, Cards, Gambling etc.. Require random Number Generators to simulate the random probabilistic events, like the throw of a dice.

Statistical Sampling and analysis: The concept of randomness laid the foundation of statistical theory, which infused the core of statistical practice. Many aspects of statistical practice were formulated on the basis of randomness and random numbers. If the statistical method is extremely sensitive to patterns in the data, very large amounts of data with no recognizable pattern are needed.

Cryptography : A pervasive use of unpredictable random numbers is in cryptography which underlies most of the schemes which attempt to provide security in modern data transfer mechanism. For eg. If a user wants to use an encryption algorithm, then it is best to use a random number as the key. If one has a pseudo-random number generator whose output is "sufficiently difficult" to predict then a true random number generated by some other means can be fed to this generator as a seed. These pseudo random numbers are called cryptographically secure random numbers.

Considering the tremendous application of random numbers in daily life, its quite obvious that it is an extensive research field. Over time computer scientists and Mathematicians have designed various algorithms that generate pseudo-random numbers. Let us have a brief analysis of the various techniques that have been developed so far.

Usually there are two major ways of generating Random Numbers:

- Hardware Random Number Generators.
- Softwares or Algorithmic Random Number generators.

**AlGORITHMIC RANDOMNESS :**

Various Mathematical techniques for generation of pseudo-random numbers include

- Blum Blum Shub
- ISAAC(Cipher)

- Linear congruential Generator
- Linear feedback shift Register
- Mersenne Twister
- Multiply-with-carry
- Xor-shift

Out of all the algorithms mentioned above, the linear congruential generator is the most commonly used random number generator and is supplied with most Programming languages. The theory behind the algorithm is easy to understand.

$X_{n+1} \equiv (aX_n + c) \pmod{m}$ , the generation is defined by recurrence relation.

where $X_n$ is the sequence of pseudorandom values, and

$m,\ 0 < m$ — the "modulus"
$a,\ 0 < a < m$ — the "multiplier"
$c,\ 0 \leq c < m$ — the "increment"
$X_0,\ 0 \leq X_0 < m$ — the "seed" or "start value"

LCGs are very fast and require very less memory ( typically 32 bits or 64 bits ). But, they cannot be used for cases where high quality randomness is critical, like in the case of monte carlo method. It cannot be used for cryptographic purpose either, because it is easily predictable since a same seed would give the same random number every time. So an attacker can detect the cipher code easily by frequency analysis. For cryptographic purposes a stronger random number generation algorithm is needed like the Blum Blum Shub. These provide a more complicated approach towards the generation of random numbers and are more efficient.

**ANALYSING RANDOM NUMBERS :**

Lets assume a completely random draw of a number from 1 to 10. So, the probability of drawing any of the numbers would be 1/10.

Now, if we assume 1000 independent draws, then probability says that each of the number should be present 1/10 * 1000 =100 times.

So mathematically, every number from 1 to 10 should be present approximately 100 times.

This condition is an important test for a random number generator algorithm, this method of testing is called the frequency analysis.

Consider the following code fragment,

```
srand(time(NULL));
for(i=0;i<1000;++i)
        array[i]=rand() %10 +1;
```

I wrote the above code as a test for the random number generator supplied with C programming language. It takes the system time as the seed and generates random numbers from 1-10.

On checking the frequency of each numbers (1-10) , in the 1000 roomed array, I found the following occurrences for each.

1 ---- 97 2 ---- 116 3 ---- 113 4 ---- 93 5 ---- 109 6 ---- 104 7 ---- 95

8 ---- 87 9 ---- 96 10 ---- 90

So, we see that it shares the statistical properties of a random number, i.e. Equiprobability of all numbers.

However, apart from the above test, for a sequence to be certified as a random sequence, a lot of other tests have to be passed. These include cumulative sums test, variance test, runs test, linear complexity test, Binary Matrix rank test etc. All these are statistically proven methods that define a random sequence.

Nevertheless, Random number generation is still an extensive research field, no algorithm has been designed till now, that can generate a truly random number. And with emerging techniques such as Monte Carlo Method, chaos theory, random-walk theory etc, random number generation, one of the strangest quests of modern computer science seems to be reaching its goal.

# Motivation

**by Pratyusa Dash,**
**Assistant Professor, CSE**

Motivation is an important concept and may be understood as the set of forces that cause people to behave in certain ways. A basic principle is that the performance of an individual depends on his/her ability backed by motivation. Mathematically it can be described as,

Performance = f (ability x motivation )

Ability refers to the skill and competence of the person to complete a given task. The person's desire to accomplish the task is also necessary.

**FRAMEWORK OF MOTIVATION**

It comprises six steps.

i. Identification of individual needs.

ii. Searching for the ways to satisfy the needs.

iii. Engaging in goal-directed behaviour.

iv. Performance.

v. Receiving either rewards or punishment.

vi. Reassessing needs deficiencies.

vii.

**IMPORTANCE OF MOTIVATION**

Motivation comprises important elements such as the need or content, search and choice of strategies, goal-directed behaviour, social comparison of rewards, reinforcement and performance-satisfaction.
The increasing attention paid towards motivation is justified because of several reasons like

- Motivated employees are always looking for better ways for do the job.
- Generally motivated employees are more quality oriented.
- Highly motivated workers are more productive than apathetic workers.
- Every organization requires human resources in addition to financial & physical resources for it to function.
- Motivation as a concept represents a highly complex phenomenon that affects and is affected by a multitude of factors in the organizational milieu.
- Motivation can be found in the present & future technology required for production.

**THEORIES OF MOTIVATION**

Several approaches to motivation are available. A perusal of the theories will help us to understand the nature of motivation better. All the theories can be classified into two broad categories- early theories and contemporary theories. Early theories include scientific management and human relations model. Contemporary theories are content, process and reinforcement categories.

**I. EARLY THEORIES**

i. Scientific Management

F. W. Taylor is known as the "Father of scientific management". Scientific management techniques includes:

Scientific method for doing work

- Planning the task
- Scientific selection
- Training and remuneration of workers

- Standardization
- Specialization & division of work
- Time & motion studies
- Mental revolution

ii. Human relations model

According to this model workers were primarily motivated by money was inadequate. Workers were expected to accept the management's authority because supervisors treated them with consideration and were attentive to their needs.

## II.CONTEMPORARY THEORIES

## A. CONTENT THEORIES

### i. Maslow's Need Hierarchy theory

This is the simplest & most widely discussed theory of motivation. This theory divides human needs into five levels like

- Physiological needs
- Safety needs
- Belongingness &love needs
- Self-esteem needs
- Self actualization needs

### ii. Herzberg's two-factor theory

This model, which is the dual factor theory and the motivation-hygiene theory, has been widely accepted by managers concerned with the problem of human-behaviour at work.

### iii. Alderfer's ERG theory

The E,R,G of ERG theory stands for existence, relatedness and growth, the three sets of needs which are the focus of the human needs in organization.

### iv. Mc Clelland's achievement theory

This is also called the three needs theory like

- Need for achievement
- Need for power
- Need for affiliation

The essence of the theory is that:

**Individual needs +responsive work environment -> Work motivation and job satisfaction**

## B.PROCESS THEORIES

### i. Expectancy theory

It is based on the idea that work effort is directed towards behaviour that people believe will lead to desired outcomes. Four important variables of this theory are

- First level &second level outcomes
- Expectancy
- Instrumentality
- Valence

ii. Equity Theory
This theory is based on the assumption that individuals are motivated by their desire to be equitably treated in their work relationships.
Four things are important in the theory like

- Person
- Comparison other
- Inputs
- Outcomes

iii. Performance-Satisfaction model
This is a comprehensive theory of motivation. According to this theory performance and satisfaction are all separate variables and relate in

ways different from what was traditionally assumed.

It can be concluded that no single theory is successful in bringing out all the complexities of motivation. All of them collectively help us to understand the behaviour of employees.

In broad sense motivation as a personal trait-that is some have it and others do not have it. In practice some students who seem to lack motivation are lazy. Motivation is the result of interaction of the student and the situation. Usually students differ in their basic motivational drive. So the concept of motivation can be analyzed to keep in mind the level of motivation varies both between students and within students at different times.

# Living one of them!

**by Ayushi Dalmia, 4th Year, CSE**

I still remember what the email said:

"Dear Applicant,

Further to your application for the internship at the Dept. of CSE, IIT Kharagpur, you are selected for the internship among other competitive applicants.

Congratulations!!! "

Reading this I was practically floating in the exosphere. A lot has happened since then but looking back I can confidently say that I picked the right place to intern with this summer.

Indian Institute of Technology, Kharagpur: yes, I got the opportunity to do an internship at one of the finest engineering colleges of the country. Like other engineering aspirants I too wished to pursue my undergraduate education from the IITs; through this internship I got the opportunity to live one of my dreams!

I worked in the Complex Network Research Group (CNeRG) at the Department of Computer Science, IIT Kharagpur where my work involved analyzing citation network[1], a type of complex network[2]. My encounter with networks and graphs began at SIGKDD, a special interest group related to Knowledge Discovery and Data Mining as part of the ACM (Association for Computing and Machinery) Student Chapter, Heritage Institute of Technology. I found it very interesting and hence I enjoyed the work here. I learnt a lot during the process, which included a better understanding of complex networks, dealing with large datasets and Python. It gave me the experience of working in the field of research, which I believe will definitely enhance my future career opportunities.

*"Dedicated to the service of the nation"*

As I entered the premises, the signage at the front of the main building displayed the above message. My first day at IIT was a flurry of activity. In the morning I had a meeting with Mr. Tanmoy Chakraborty, PhD. Scholar at CNeRG and my mentor for this summer. In the meeting I figured out what I would start working on after which I spent the day getting the computer assigned in the software lab, getting hostel accommodations (which again was an entirely new experience) and other errands. I remember feeling completely overwhelmed; I was thrown head over heels to a completely new place and I wasn't familiar with handling large datasets. But my mentor guided me so well and was always willing to answer any question that I had. With time I gained confidence and started enjoying my work. I also started getting comfortable with the place and as I began to feel like home, my productivity also went up.
Getting into the technical details of the project, my work involved analyzing two problems:

- **Overlapping Community Detection in Citation Networks:**

The structural modularity and compartmentalization of a complex network is closely related to the dynamics towards clustering. Most of the community detection algorithms are inefficient at capturing overlaps in-between communities, detecting communities having disparities in size and density, and taking into account the modular structure of multipartite networks. In this experiment, for the first time, citation network has been visualized as a tripartite hypergraph[3]. We detect the overlapping community structure simultaneously from the three partitions. Our algorithm modularizes the hyper-edges using unipartite community detection algorithm after converting citation hypergraph into its corresponding weighted line-graph, and hereby produces the overlapping communities of three partitions. We aim to illustrate its efficiency through extensive experiments on synthetic as well as large scale real citation data of computer science, and then compare it with existing state-of-the art unipartite overlapping community detection algorithm performing on dynamic networks.

❖ **Analysing and Comparing Citation Network after removing self citations**

In the field of research and analysis the importance of a paper is heavily dependent on the number of citations received by the paper. Authors, in general have the tendency to cite those papers, which are written by them or one of their collaborators. Doing this increases the impact factor of papers. Also, if we consider the most cited author it happens that the popularity of the author increases due to these self-citations and collaboration citations. In this work we studied and delved into answering several questions pertaining to this issue.

In the brief period of 2 months I had many experiences that are worth mentioning. Besides the work, I made a lot of friends amongst the other interns. We watched movies, lunched together every day and checked out local joints. All the fun, the learning and the challenges made time fly by and before I knew it, my last day as an intern had arrived. The last day was hectic (in contrast to my first). I had to finish up some paper work, submit my projects and of course, have fun because that might just have been my last day there. I had one last dinner with them that night, friends and seniors all together, and then it was back to home.

***Notes:***

1) Citation Networks: A citation is a type of complex network of citations between publications, allowing the user to easily establish which later documents cite which earlier documents.

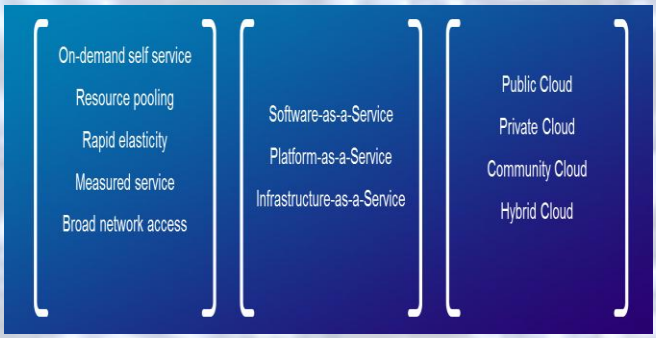
2) Complex Networks: In the context of network theory, a complex network is a graph (network) with non-trivial topological features—features that do not occur in simple networks but often occur in real graphs.

3) Hypergraph: A hypergraph is a generalization of a graph in which an edge can connect any number of vertices

# A Cloud in a box

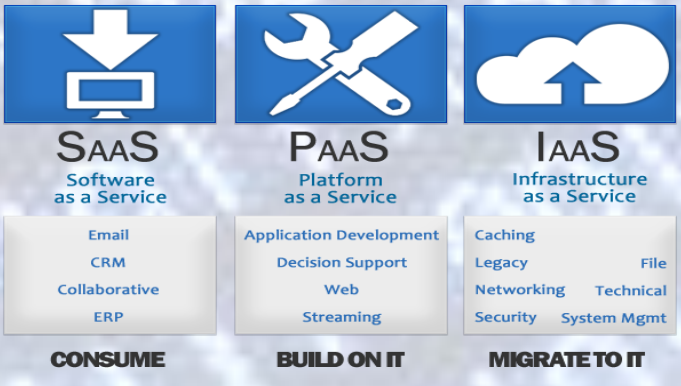
**by Ranadeep Guha, 4th Year, CSE**

Cloud computing is an emerging technology in the world of computing, that delivers computing resources as a service over a network. It is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential **characteristics,** three **service models**, and four **deployment models**.




There are three types of cloud computing:

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS), and
- Software as a Service (SaaS).




Using **Infrastructure as a Service**, users rent use of servers (as many as needed during the rental period) provided by one or more cloud providers.

Using **Platform as a Service**, users rent use of servers and the system software to use in them.

Using **Software as a Service**, users also rent application software and databases. The cloud providers manage the infrastructure and platforms on which the applications run.

In the summer of 2012, I set out to implement Infrastructure as a Service as a part of the summer project under Ericsson. With my project partner Mayuresh Harsha and the assistance of our project mentors-Mr.Sanjib Mallick (Head of Cloud Computing), Mr.Sriraj Muralidharan (Senior Solution Integrator), Mr. Dipayan Sanyal (General Manager) , we managed to implement Infrastructure as a Service on our computers with the help of Virtual Machines.

The Virtual Machines are spun off using an interactive interactive menu driven shell script program so that the end user can get the necessary infrastructure according to his/her needs and to create a virtual network between those virtual machines through network bridges. In order for guests to appear as individual and independent systems on the external network (i.e. with their own IP addresses), they must be configured to share a physical network interface on the host. This is achieved by configuring a network bridge interface on the host system to which the guests can connect. It should be kept in mind, that this project could be used in a larger resource pool without making much change.

Some of the common terms associated with this project are as follows:

## Virtualization

In full virtualization, a layer exists between the virtualized operating systems and the hardware as a way to arbitrate access. This layer is called a **hypervisor**, or virtual machine monitor (VMM). At the bottom of a virtualization solution is the machine to be virtualized. This machine may or may not support virtualization directly, which then requires support by the next layer, called the hypervisor. The hypervisor, or VMM, serves as an

abstraction between the platform hardware and the operating systems. In some cases, the hypervisor is an operating system. We will use KVM as our hypervisor.

## KVM

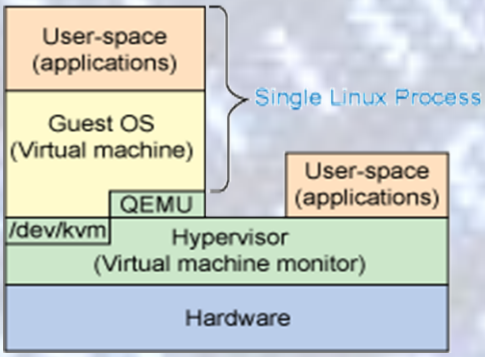
KVM (for Kernel-based Virtual Machine) is an open-source software providing full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V).
It consists of a loadable kernel module, kvm.ko, that provides the core virtualization infrastructure and a processor specific module, kvm-intel.ko or kvm-amd.ko. Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

## QEMU

QEMU is a generic and open source machine emulator and virtualizer.

When used as a machine emulator, QEMU can run OSes and programs made for one machine on a different machine (e.g. your own PC). By using dynamic translation, it achieves very good performance.

When used as a virtualizer, QEMU achieves near native performances by executing the guest code directly on the host CPU. QEMU supports virtualization when using the KVM kernel module in Linux.




The above diagram shows a view of virtualization with KVM.

## Command Line Interface (CLI)

A CLI (command line interface) is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth. The MS-DOS Prompt application in a Windows operating system is an example of the provision of a command line interface.
The following goals needs to be achieved-

- **On demand service**- This is achieved by provisioning an automated spinning of Virtual Machines.
- **Resource Pooling**- This is achieved by adding storage elasticity and network capabilities.

**Measured Service**- This is achieved by utilization report of the various resources used

**SETTING UP THE MACHINE FOR THE PROJECT**

1. Check in the BIOS if virtualization is supported. For someone who uses an Intel processor, he/she can use the Intel® Processor Identification Utility software to check it as well.
2. Install Fedora (which was used in this project) or Ubuntu on the system.
3. Install KVM from Fedora.
4. Run the shell script.
5. Configure the bridged network.

Ping two virtual machines


**To install KVM**
**#su –c "yum install @virtualization"**
**To verify KVM**
**#lsmod | grep kvm**
This displays-
kvm
kvm_intel

**CREATING THE MENU DRIVEN SHELL SCRIPT**
**Displaying a menu**

Display a menu, where the user gets an opportunity to install a virtual machine, check status of an already running virtual machine or shut down an already running virtual machine. The menu is kept inside an infinite loop.

```
while [ 1 -gt 0 ]
do
echo "press 1 to install virtual machine"
echo "press 2 to get status"
echo "press 3 to terminate"
read a
```

**LIBVIRT**
Libvirt is the virtualization API which provides a common layer of abstraction and control for virtual machines deployed within many different hypervisors. The main components of libvirt are a control daemon, a stable C language API and a corresponding set of Python language bindings, as well as a shell environment.

**VIRSH**
Virsh is a console program maintained by the Virt-Manager team. It provides an easy to use console interface to the libvirt library for controlling guest instances. Each of the commands available within virsh can be used either from within the virsh environment itself, or called from a standard Linux console.

**Installation a Virtual Machine**

In order to install a virtual machine, virt-install must be run as root and accepts a wide range of command-line arguments that are used to provide configuration information related to the virtual machine being created.

Let us take an example of a virtual machine, which we need to spin off.

Let the configuration be- 2 CPUs, 250 megabytes of RAM and storage of 8 gigabytes.

**Install a Virtual Machine**

The code snippet to install this virtual machine is-

```
virt-install --connect qemu:///system -n fedora1 -r 250 --vcpus=2 --disk path=/var/lib/libvirt/images/fedora1.img,size=8 -c /var/lib/libvirt/images/fedora17.iso --vnc --noautoconsole --os-type linux --os-variant fedora13 --accelerate --network network=default --hvm
```

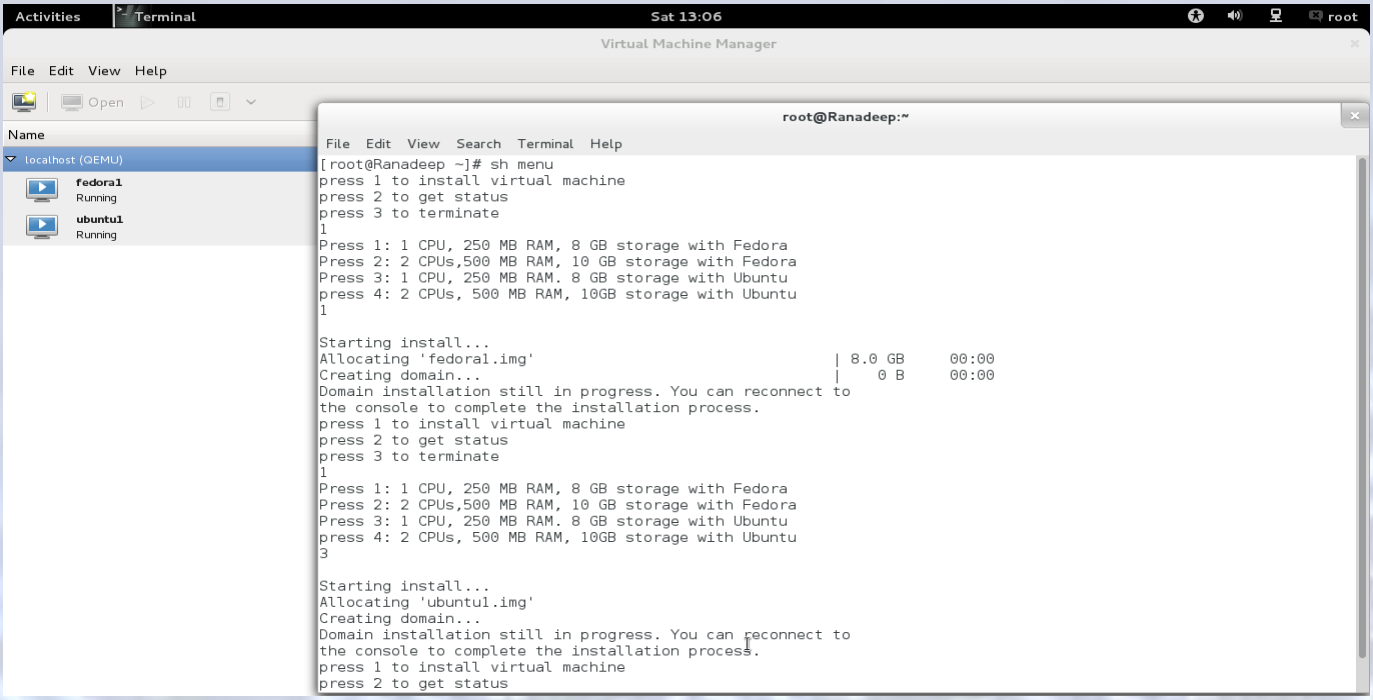
This uses the .iso file located in the path /var/lib/libvirt/images to create a guest with the name fedora1, 250 megabytes, 2 virtual CPUs and the disk image file named fedora1.img.

**Get status of Virtual Machine**

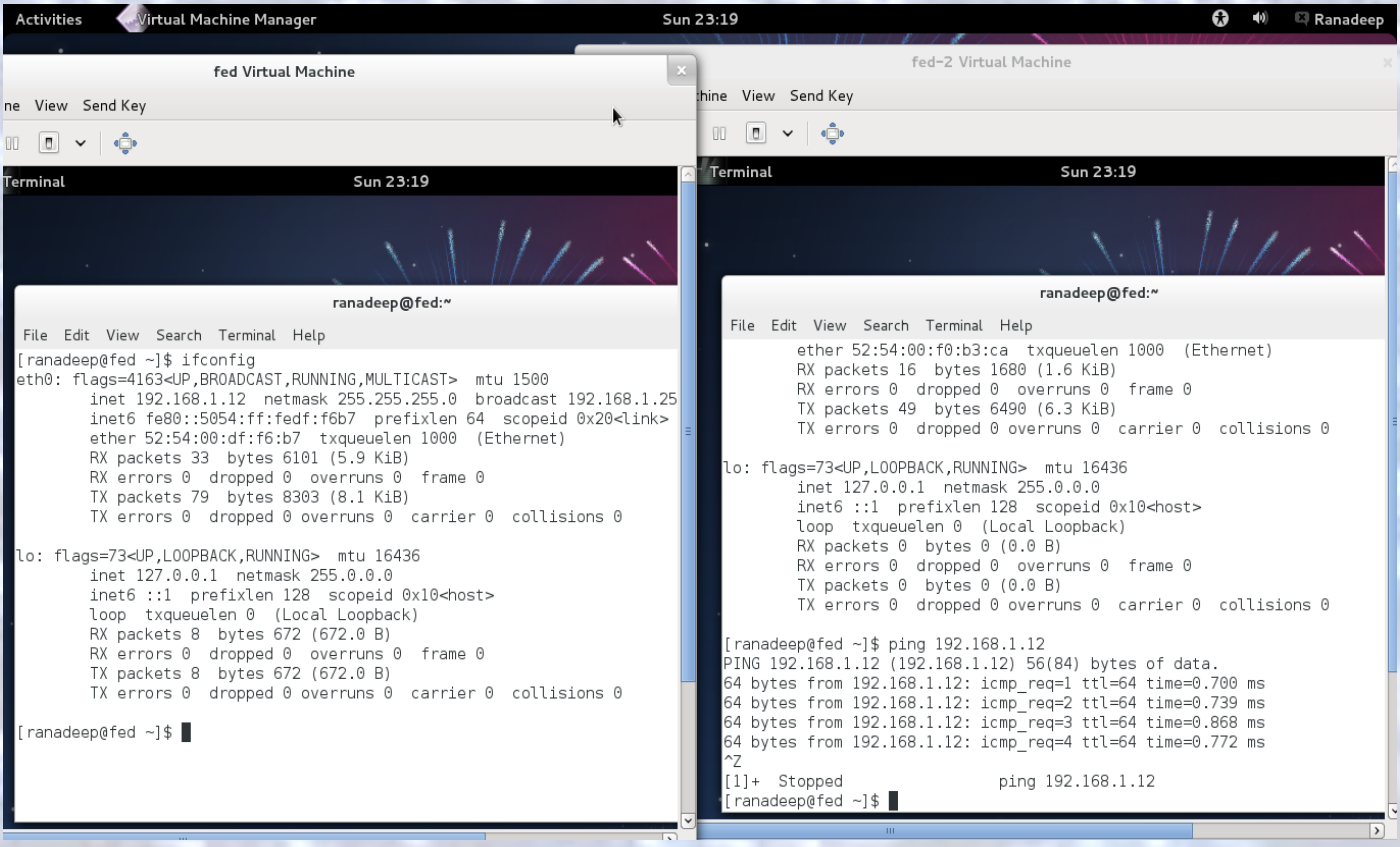
```
virsh dominfo fedora1
```

**To shutdown Virtual Machine**

```
virsh shutdown fedora1
```




The above diagram shows installation of 2 virtual machines fedora1 and ubuntu1 using the shell script.




The above diagram shows 2 Virtual Machines pinging each other acting as independent machines.

# Magic and Computer Science

**by Md Aman Khan, 3rd Year, CSE**

Note :- Some of the materials for this article have been derived from several books , google images , internet sites , and tricks performed by real magicians

The world that we see around us is full of magic , co-incidents and supernatural phenomenon which seem unexplainable for a common human being . However not all of these events are really unexplainable. Let us consider the tricks performed by a magician. A layman feels that the tricks are

impossible and in fact the tricks are designed for us to feel that way. However in reality the tricks are mere use of sleight of hand and a little bit use of algorithm ( which we can even apply on a computer ) . The tricks which require sleight of hand may have a 99 % chance to work however the time when we try to work out the trick in front of our friends or a large audience might turn out to be that 1% (doomsday for the magician) . Thus we will here be concentrating on those magic tricks (card tricks) which are bound to work if we strictly apply the algorithm designed for it. Illustrations via images have been given wherever it seemed necessary and appropriate. The computer science link is to something deeper than today's technology, something Fundamental in the subject. The second section of each trick describes this Computer Science link. We hope you will find the science and Maths as fascinating as the tricks

## LET THE MAGIC BEGIN

### THE OUT-OF-BODY EXPERIENCE :- THE ONE WHERE YOU FLOAT OUT OF YOUR BODY TO WATCH EVENTS

**Trick Description**

You are blindfolded and lean against the wall at the back of the room with your back to the proceedings. Your spirit leaves your body and flies up to the ceiling so you can watch from above. Meanwhile, your assistant shuffles a pack of cards. Volunteers then select cards and place them at random either face-up or face-down in a 4 by 4 grid. Your assistant adds more to make it even harder. Your spirit now has a target to watch. A further volunteer then chooses any card from the grid and flips it over. No-one speaks. You are still blindfolded. You can only know which one was flipped if your spirit really is floating above, watching. You are told to return to your body, which you do. A little dazed, you go straight to the cards
and point to the one that was flipped over!.

**Trick mechanics**

This trick is a flamboyant variation of one invented by New Zealand computer scientist, Tim Bell. Have a look at the set-up in the diagrams below. You might just catch the workings of the trick.
The assistant adds an extra row and column of cards but it isn't in fact random. It also isn't actually making things harder, but easier. What they do is look at the number of facedown cards in the row /column(the number of card backs showing) and if that number is odd, they put the new card face-down else face up . Repeat this for all the rows, then do the same for the columns. Add the extra card so that there is an even number of card backs in the column/row. The final card on the bottom right of the last row finishes the set.

Detecting the change doesn't now need any special mystical abilities. it's a simple process to find its location shown in the demonstration. Look back at the cards. Start from the top, scanning down row-by-row looking for card backs. Remember you added the extra cards to ensure there was an EVEN number of backs in the row. There will be one row where there are now an ODD number of backs; one of the cards in this row was turned over, but which one? Start to scan the columns now, again looking for the column where there is an ODD number of card backs showing. When you find it that's the column with the reversed card. So you now have the row position and the column position of the reversed card, and you can reveal this in any super memory sort of way you like.


Detecting the flipped card using parity

## A demonstration

4 by 4 grid of random cards laid out by the volunteer

Assistant adds an extra bit of rows and columns to make it even harder

**TRICK ALGORITHM**

**1** Blindfold yourself

**2** Select volunteer

**3** Ask volunteer or assistant to shuffle the pack of cards.

**4** Ask volunteer to choose cards randomly and place them on a 4 by 4 grid with face up or down randomly.

**5** Assistant adds an extra row of cards and an extra column of cards and whether the face of card is up or down depends on parity (face up if the number of face down cards in row/column is even else face down) .

**6** Ask another volunteer to flip one of the cards.

**7** Depending on the parity row find out the row which has odd number of face down cards and the column which has odd number of face down cards . The intersection gives the flipped card,

**8** Announce the result

**RELATIONSHIP WITH COMPUTER SCIENCE**

## Finding mistakes in data – parity

What does this trick have to do with computer science? In the figure the extra row and column you add have a technical name: the **'parity'** row and the parity column. (Parity means equal). Instead of thinking about face-up and facedown cards, think about binary 1 and 0. You can see that your block of cards could just as easily represent a segment of computer data, with the data encoded in 1's and 0's. (These are called **'binary bits'**).

Data sent over a computer network is just a series of 1s and 0s (each called a **'bit'**) packaged into blocks. Trouble is the real world is a 'noisy' place. Signals can be corrupted in all sorts of ways: cosmic rays, radio signals, nearby power lines and the like can all zap bits. It's easy for them to be flipped as they pass over a network. One change can destroy the whole meaning of the message.

To ensure that, when you send data over a computer network, all the data does make it to the other end without getting scrambled, computer scientists and engineers came up with the idea of adding parity bits to each block of data. It is no different to the way you added the extra cards. Suppose you want to send a message over a network consisting of the numbers 6, 13, 2 and 12. They can be converted into binary using a special code where each number has its own sequence of 1s and 0s to represent it (see page 17). Our numbers are converted to the four sets of digits: 0110 1101 0010 1100. Rather than send those digits though we add the parity bits to make them five digits long with an extra block at the end for the column parity:

**01101 11010 00100 11001 10101**

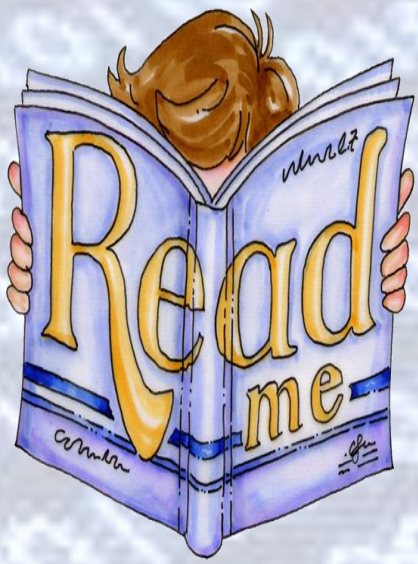
We have used the parity bits to give an even number of zeros here. Now when the data arrives the receiving computer can see if one of the bits (cards) has an error e.g. it's 1 when it should be 0 or vice versa. Suppose the computer at the other end actually receives the following message:

**01101 10010 00100 11001 10101**

By lining the separate groups back into a rectangle, we can see where the parity has been broken in row 2 and column 2 as they both now have three 0s whereas everything else is still even:

**01101 10010 00100 11001 10101**

## THE ONE WHERE YOU READ PEOPLES' MINDS LIKE BOOKS : -




**Trick Description(21 cards involved)**

Call a volunteer to shuffle a pack of cards.
Now deal out single cards, left to right into three piles of seven cards, all face up and visible and leave out the rest f the cards . The volunteer mentally selects one of the cards. Now it is your job to read their mind and tell them the card they are thinking of...
This of course does not seem too easy unless the volunteer is a part of the trick and you know from beforehand which card he is going to choose . However that is not the case here .

The volunteer should not tell you which card he/she has chosen but you should get them to tell you the pile it is in. (probability of you guessing the card is 1/7 then since each pile has seven cards) . You collect up the cards, and deal them out a card at a time left to right into three piles once more. Again they tell you the pile their card is in, you collect the cards once more, saying that they are tough to read. Deal the cards out across the table in the three piles again in the same way. Your friend indicates the

pile their card is in. Collect the cards again and deal them into the three piles one last time. You immediately announce their card and magically it is in the very middle position of the pack

**Trick mechanics**

The real trick does not depend on any psychic powers at all . In fact , it is dependent on the cunning way of shuffling and dealing the cards . The trick involves several deals each apparently shuffling the order of the cards but the simple thing that you need to do is always put the pile that your volunteer selected in between the other two piles . If we do that then the fourth deal will ensure that the card the volunteer selected is exactly in the middle of the pack ( always ) . This happens because the positions in which the chosen pile cards can go is limited as will be visible from the given demonstration . Thus it does not matter which card is chosen it will always land up in the middle of the pile

A demonstration

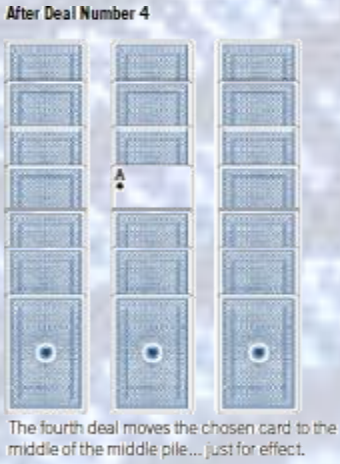
After deal 1 say
The volunteer had the black ace of the middle pile in mind

The cards of the middle pile are distributed throughout the piles



The choices for the cards is reduced and we are left with three cards which perhaps the volunteer had chosen

The card that had been chosen by the volunteer has reached its final position ( middle of the pile )



**TRICK ALGORITHM**

**1** Select volunteer

**2** Ask volunteer to shuffle the pack of cards.

**3** Deal out single cards, left to right into three piles of seven cards, all face up and visible.

**4** Ask volunteer to choose a card in mind.

**5** Ask volunteer to indicate the pile the card is in.

**6** Put the pile the volunteer chose in middle of the other two piles

**7** Deal out single cards, left to right into three piles of seven cards, all face up and visible.

**8** Repeat steps 5 to 7 two more times.

**9** Announce that the card in the middle of the pack , that is the 4th card of the second pile is the card the volunteer chose

**RELATIONSHIP WITH COMPUTER SCIENCE**

The relationship of this trick with computer science is not so evident but there definitely is one crucial thing that is in direct relation with the science of computers and that is testing . If we are required to test the algorithm for all possible inputs then it will seem that there can be 21! Ways of arranging the 21 cards or perhaps even more since the choosing of the 21 cards also needs to be considered . However that is not the case here . The testing of the algorithm can be done by using 21 choices as there are 21 initial positions which the volunteer may have chosen .

We come to this conclusion because the final result is independent of the cards which had not been chosen . They do not affect the result in any possible way . thus it does not matter how the arrangement of the other cars was and in fact it does not even matter which was the card that was chosen . The only thing that matters is the position of the card that was chosen . This kind of simplification is always observed I computer science programming I order to reduce complexities ( time and space ) and make the algorithms more efficient and thus this testing procedure is in direct link up with computer science .

**ARE YOU PSYCHIC ???**

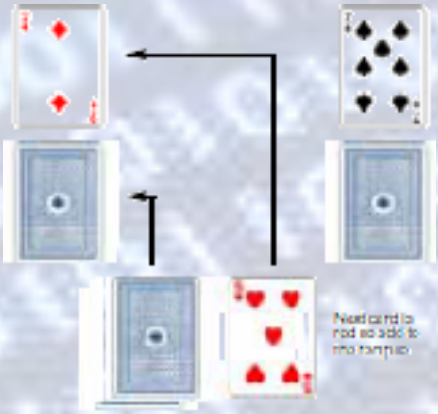
**Trick Description**

Get a deck of cards and give them a good shuffle. Spread the cards on the table face down. Now think of the colour RED and select any eight cards, then think of the colour BLACK and select another seven cards at random. Now think of RED again, select another six random cards then finally BLACK again and select five cards. Shuffle the cards you chose, and then turn the pile face-up. Take the remaining cards, shuffle them and spread them face down. Now the remote control starts. Concentrate. You are going to separate the cards you selected (and that are now in your face-up pile) into two piles: a RED pile and a BLACK pile, in the following way.

Go through your face-up cards one at a time. If the next card is RED put it in the RED pile. For each RED card you put in your RED pile think RED and select a random card from the face down cards on the table without looking at it. Put this random card in a pile face down in front of your RED pile. Similarly if the next card is a BLACK card put it face up on your BLACK pile, think BLACK and select a random face down card. Put this face- down

card in a pile in front of your BLACK pile. Go through this Procedure until you run out of face-up cards.

You now have the following: a RED pile and in front of that a pile containing the same number of face down cards you selected while thinking RED. You also have a BLACK pile in front of which is a pile of random cards you selected while thinking BLACK. Interestingly your thoughts have influenced your choice of random cards! Don't believe me? Look at the pile of random cards you chose and put in front of your RED pile. Count the number of RED cards in this pile. Now look at the random cards in front of your BLACK pile, and count the number of BLACK cards you selected.
You selected the same number of RED and BLACK cards totally at random! -One card out and it wouldn't have worked! It's a final proof that your sub-conscious mind can make you choose random cards to balance those numbers! ... Or is it?



**TRICK ALGORITHM**

**1** Select volunteer.

**2** Ask volunteer to shuffle the pack of cards.

**3** Spread the cards face down on the table.

**4** Ask volunteer to think of red and choose 8 cards randomly.

**5** Ask volunteer to think of black and choose 7 cards randomly.

**6** Ask volunteer to think of red and choose 6 cards randomly.

**7** Ask volunteer to think of black and choose 5 cards randomly.

**8** Shuffle the selected cards and place the pile face up

**9** Shuffle the non selected cards and spread them face down.

**10** If card on face up pile is Red place the card in a separate pile (say named Red Pile ) and choose a random card from the face down cards and place in front of the Red pile .

**11** If card on face up pile is Black place the card in a separate pile (say named Black Pile ) and choose a random card from the face down cards and place in front of the Black pile.

**12** Repeat steps 11 and 12 until all the cards of the face up pile are finished

**13**. Reveal that the number of red cards inn the pile in front of red pile = number of black cards in pile in front of black pile .

**TRICK MECHANICS AND RELATION WITH COMPUTER SCIENCE**

Maths is a very important part of computer science . All programs and algorithms heavy rely on maths to obtain the objectives and maths has magical properties . This magic trick is just an example to show the magical effects of maths , more specifically algebra .


R1- ony red cards of red pile
B2- only black cards of black pile

R3 – red cards of the pile in front of red pile
B3 – black cards of the pile in front of red pile
R4 – red cards of the pile in front of black pile
B4 – black cards of the pile in front of black pile.

**R1 + R3 + R4 = 26** Call this equation (1)
**B2 + B3 + B4 = 26** Call this equation (2)
We also know the number of cards in the RED pile 1 (R1) is the same as the number of face down cards placed in front of it in pile 3 (made up of R3 red cards and B3 black cards) so together R3+B3 must add up to R1. Similar reasoning holds for the cards in front of the BLACK pile (pile 2 with pile 4). So we know two more equations:
**R1 = R3 + B3** Call this equation (3)
**B2 = R4 + B4** Call this equation (4)
Now using 1 and 3
**(R3 + B3) + R3 + R4 = 26** Call this equation (5)
Similarly if we substitute equation (4) in equation (2) eliminating B2 we get
**(R4 + B4) + B3 + B4 = 26** Call this equation (6)
Combining equations (5) and (6) as both add up to 26, we get
**(R3 + B3) + R3 + R4 = 26 = (R4 + B4) + B3 + B4**
We can simplify this by grouping the same things together
**2xR3 + B3 + R4 = R4 + 2xB4 + B3**
**2 x R3 = 2 x B4**
**:R3 = B4**
Now what did we say R3 and B4 stood for? They are just numbers of cards of particular colours in the face down piles.

**CONCLUSION**

Thus the importance of computer science and maths is evident in all fields and also in a major way in the filed of magic as has been proved by the above tricks . There are several other tricks that are based upon maths and algorithms giving 100% results and having direct relationship with computer science . However due to lack of time and trying to keep the article short enough to read we have only kept three tricks . However we may reveal more tricks in our future articles . Hoping that you liked the article

**Sources for images and material :-**


1 http://www.sylviasierhuis.nl/evenementen/concepten/

2 www.paradigmshiftcentral.com/datamine/astral-projection.html

3 http://anythingeverythinghere.com/the-different-types-of-psychic-readings/

4 cs4funmagicbook 1 and 2

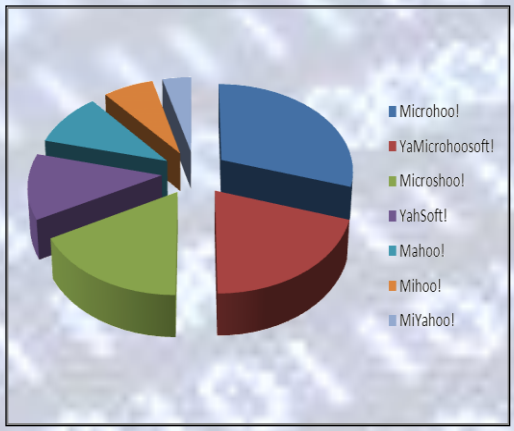
# MicroHOO! CARES!

**by Nilina Bera, Assistant Professor, CSE**


What's all about this freaky title MS + Yahoo! = Microhoo!

It's Official! Finally The! people! have! spoken!

With the new logo of the joint venture, the much awaited buy-off between Microsoft getting the Yahoo Search Engine under its hood has ended with people's vote:




Well, one fine chilly morning, suddenly if we get to know about the news "**Microsoft has moved to buy *Yahoo!***" for $44 billion!!!, a series of questions start hitting our minds:

1. What does a combined Microsoft *Yahoo!* Look like?
2. What 'Microhoo' means for Google?
3. What would happen *to cloud computing* space as both Microsoft and *Yahoo!* are competing vendors?
4. What this means for users and lovers of technology?
5. Is *Yahoo!* by Microsoft real?

Are we watching this humongous deal as reality show?

Before proceeding further, some key-terms that would assist us to dive deep into Microhoo! -- the combination further generating more technical terms like

- Microsoft,
- Yahoo!
- Cloud computing
- Search Engines

…the hot components mentioned above ironically producing the hot-creamy-chocolate shake and we are NOT kidding over here!!!

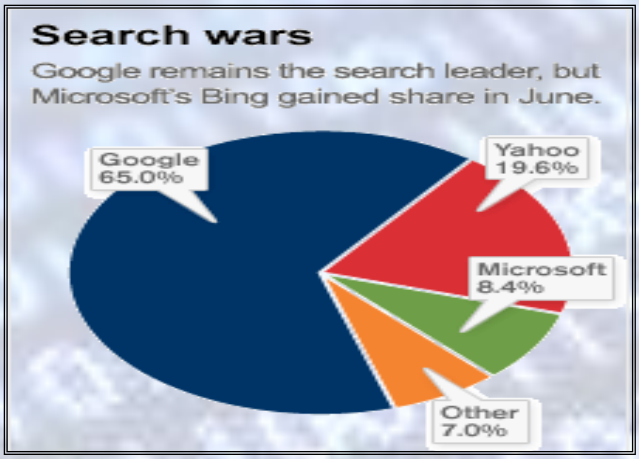
## What 'Micro*Hoo*' means for Google? 'Micro*Hoo*' vs Google 2014

Analysts are divided on whether the Microsoft and Yahoo combination can hurt Google, but most agree that Google is safe in its top spot. "Microhoo" is finally a done deal, but will it really be able to make a dent in Google's enormous search market lead? Google maintains 65% of the U.S. search market, compared to a combined 28% for Microsoft and Yahoo. But the newly partnered tech giants are hoping that ***one plus one equals more than two.***

"This deal is really about scale," said Yahoo (YHOO, Fortune 500) Chief Executive Carol Bartz on a conference call. "By combining the ... technology of both companies, we can create a real, viable alternative for advertisers."

In other words, size really matters to advertisers in the search market. More data means more relevant searches and ads, which means more money can be charged to advertisers.

**Not No. 1, but strong No. 2**. Even if Microhoo doesn't make a run at the top of the search market, some argue that the deal is good for firms, advertisers and consumers.




## What are 'Cloud' and 'Cloud Computing'?

The term "*Cloud*" is used to describe the Internet in a sense different from its literal meaning. The metaphor is based on how the Internet is described in computer network diagrams.

It is difficult to precisely define the term "*Cloud Computing*" because it is an emerging technology and is rapidly evolving, unfolding itself in multidimensional real-world applications.

The most widely accepted definition is "*Cloud Computing is a pattern (paradigm) of computing in which dynamically scalable and virtualized resources are provided as a service over the Internet*".

## Role of Micro*Hoo!* in a Cloud…

In 1999, Salesforce.com began operation applying technologies developed by Google and Yahoo to business applications. Business users widely welcomed the services because of their *flexibility*, *speed* and more importantly, *customizability* by *customers* themselves.

In 2000, Microsoft extended the concept of SaaS (Software as a Service) through improved Web Services. IBM improved the concept introducing *self-monitoring*, *self-healing*, *self-configuring* and *self-optimizing*[1] in the management of IT systems.

In 2009, Cloud Computing solutions by Microsoft, *Yahoo!*, Google, Amazon, IBM, Ubuntu and Sun moved in the high demand zone and in the ushering of 2013 the demand is still *ON*!

## Micro*Hoo*! What does it mean for Users?

User Community votes ➔ **Good news for us!!!** ➔ **Why the users think that the news is *good*?**

It's going to validate a lot of innovation at Yahoo! Microsoft is focusing on what this acquisition means for advertising and for search. Since when is Yahoo! particularly good at either of those things, though? Yahoo! has created a web presence with more traffic than almost anyone else on earth. That's what they are good at and the issue is that they haven't been able to make money off of it.

Yahoo! is great at content and online innovation, though. That's what Microsoft needs right now. Google is posing a threat to Microsoft not just because it is winning in advertising, where Microsoft is a relative beginner, but because Google is shifting the software world to online.

Microsoft is serious about innovation; they just haven't been doing much of it in house for a while. The Live.com work and the Microsoft acquisitions in the health space indicate to us that the company

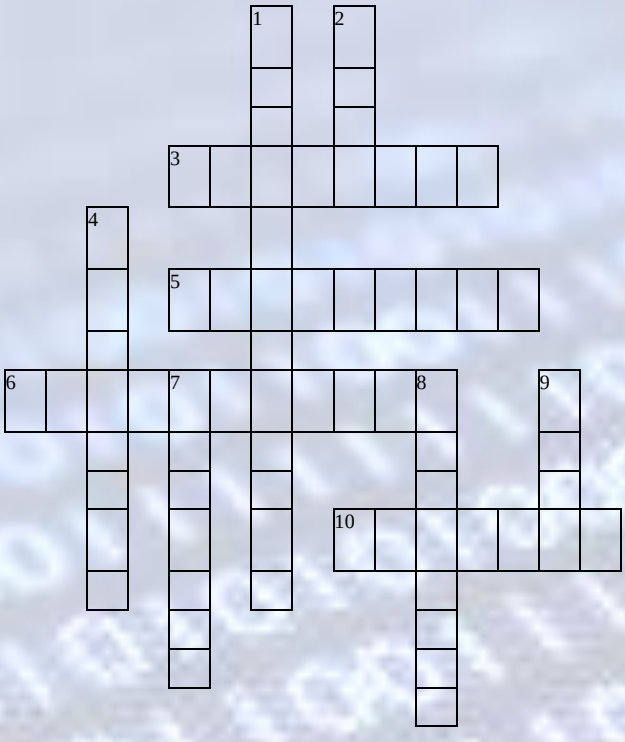
[1] The terms (self-***) are the determinants of scalability.

really is trying to do more than just to catch up the share in search and advertising domains.

By considering the relevant facts, it seems that this acquisition meant a whole lot more energy put behind **services** like ***Flickr*** and ***Del.icio.us*** and innovative **content sites** like ***Yahoo! Sports*** and ***Yahoo! Finance***. ***All of that will be good for Microsoft and it will be good for those of us who find those sites and services inspiring.***

# CROSSWORD

(***Based on ACM***)



**Down**

**1.** First woman to win ACM Turing Award(7,5)

**2.** Annual International Programming Contest held by ACM(4)

**4.** First person of Indian origin to receive the ACM Turing Award(3,5)

**7.** Head quarter of ACM is located at _____(3,4)

**8.** IBM computer to beat World Chess Champion in 1997(4,4)

**9.** ACM project (2004) for faculty supported by grants from The Andrew W. Mellon Foundation(4)

**Across**

**3.** The present president of ACM (2012 - 2014)(4,4)

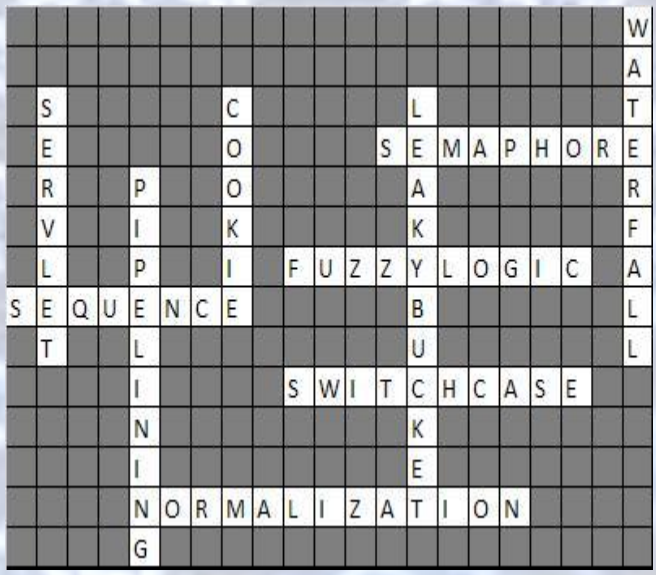
**5.** Century of formation of ACM(9)

**6.** ACM's annual prize recognized as Nobel Prize of Computing(6,5)

**10.** Precursor of the Internet(7)

***Solution will be published in the next issue. You can also send in your answers to hitkacm@gmail.com***

***Solution for the crossword published in the last edition:***




***You too can contribute! Send in your article or feedback or letter to the editor to reshma.roychoudhuri@heritageit.edu***

## Solution

[to the Puzzle mentioned in the article **"Have you ever thought what others are thinking"** by Dr. Subhashis Majumder, Professor and HOD, CSE, in HITech June 2012 issue.]

Consider that there was only one squint-eyed person in that village (number of villagers does not matter). Then try to think what these villagers will think. The squint-eyed person will see that everybody else is not squinted. Then she also knows that there is at least one squint-eyed person in the village and it must be she herself. So she will come out at the end of the 1st day. Note that every other villager will see that there is already one squint-eyed person they can see, but none will be sure about her own status. So they will remain in the queue.

Now consider, if there were two squint-eyed persons in the village A and B. A will see that there is one squint-eyed person, but since she will not be sure about herself at the end of day one, she will not come out. The observation and thinking will be exactly same for B. However for everybody else, she will see that there are two squint-eyed persons already and none of them will be sure about her own status at the end of day one. However, on the second day, A will think why did not B go out at the end of day one, then 'I must be squint-eyed'. Hence at the end of day two, A will go out of the queue. The thinking of B will be exactly like A and with a similar reasoning she will also be sure about herself and come out of the queue. Everybody else will still not be sure about herself and hence will not move.

If we now extend this concept it is quite obvious that if there are 40 squint-eyed persons, none will be sure till the 39th day, whereas on the 40th day, each one of the squint-eyed person will think – I can see 39 squint-eyed persons, why did not any of them go out last evening. 'Then I must be also squint-eyed'. Hence at the end of the 40th day all these 40 people will come out as a group