

CSE NEWSLETTER

BITS & BYTES

8TH
EDITION

COVER STORY
REENGINEERING
REMBRANDT

INSIDE
GIZMO TALK
VOICE STICK

ALGORITHMS
HASHING

CROSSWORD

COMPUTER SCIENCE AND ENGINEERING, NIT TRICHY.

TABLE **OF CONTENTS**

EDITORIAL 01

COMPUTER APPLICATION 02

ALUMNI INTERVIEW 04

PRACTICAL PROBLEM 06

GIZMO TALK 09

RECENT EVENTS 11

ALGORITHMS 12

CROSSWORD 14



EDITORIAL

HEAD OF DEPARTMENT

Dr. Leela Veluswamy

FACULTY ADVISOR COMMITTEE

Dr. Leela Veluswamy

Prof. R. Gururaj

CHIEF EDITOR

Akhila Yerukola

CO- EDITOR

Shruthi Muthukumaran

Prasann Dubey

CONTENT HEADS

Ashutosh Agarwal

Jayti Singh

DESIGN HEAD

Anurag Kumar

EDITORS

Aditya Balaji

Sirisha Konathala

Anugrahaa Ramesh

Keerthana Ganesan

R G Thivyavignesh

CONTENT TEAM

Nandhitha Raghuram

Siddharth Elango

Arvindmani

Ayush

Rahul Kar

Panigrahi

Parth

Sachin

Soorya N

DESIGN

Shilpa

Pradhiksha

Lakshmanaram

Manjith

Ritul

Amongst the plethora of course options available at NIT Trichy, the Department of Computer Science & Engineering offers one of the most sought after courses. With the freshly revamped curriculum and Honours option, the department is making strides in the direction of delivering a class apart education.

In this pursuit, we explore various co-curricular possibilities to enhance student exposure to the field of computer science. Vortex, the annual symposium held in the month of January aims at providing a platform to give students both Industrial and competitive coding experience. With industrial experts such as Sanket Saurav – founder of DoSelect, Krishma Singal – associated with IBM Watson project and many more, gracing the Guest Lectures dais, the students gain a unique insider perspective of the Corporate World. Events such as Paper Presentation, various workshops and the student populace that annually participates in Vortex, overall present students with a high class learning experience.

As the curtains are drawn on this academic year, Bits & Bytes hopes to provide a final glimpse of CSE and its varied branches and applications through interviews, research avenues, gizmo talks and much more.

We sign off hoping that we've been able to make a mark in your academic experience and will continue to do so in the forthcoming editions.

- Shruthi Muthukumaran

COMPUTER APPLICATIONS

RE-ENGINEERING REMBRANDT

SACHIN ASHOK
SOORYA NARAYAN



If bringing alive a 17th century artist wasn't something you were expecting computer scientists to be doing with their time, think again.

Don't jump to any conclusions regarding the great Dutch etcher – Rembrandt van Rijn, being brought back to life. That's up to the biologists.

What we do have though, is a weird new painting.

Funded by ING and backed with technology from Microsoft, a group of scientists with a 3D program have created a new Rembrandt masterpiece by infusing hundred and forty-eight million pixels with more than a hundred and sixty-eight thousand data points.

Well, the real Rembrandt wasn't around for that, so they had to crunch a lot of numbers to achieve what they did.

Confused?

As the popular saying goes, with great number crunching ability, comes, great responsibility. And with all the number crunching that was happening, someone was bound to find something interesting.

By analyzing the entire range of the master's work, pixel by pixel, using high resolution 3D scans, the team was successful in employing their deep learning algorithms on the 150GB of data collected, to find artistic patterns in the master's work.

Every painting needs a subject and Rembrandt's favourite, was his fellow humans.

The researchers narrowed down their analysis to Rembrandt's portraits and began to determine the common elements in his paintings.

Breaking down the process, starting with gender and age, to the amount of facial hair present, the data led them to a conclusive subject – a portrait of a Caucasian male with facial hair, aged between thirty and forty, adorned in black clothes with a white collar and a hat, facing to the right. With the subject now determined, the team had the simple task of replicating the unique style of Rembrandt. It's hard enough to copy an existing painting, but to create one out of nothing but numbers, such that Rembrandt wouldn't be left rolling over in his grave, requires utmost skill and persistence.

A great deal of attention was given to the sharpness of images and the lighting and shadow; facets so important to his work that they have their own Wikipedia entry. To put that in perspective, most painters almost never get that privilege.

With all the grub work out of the way, the only thing left to do was to paint the picture.

To go about doing that, they created a height map using two different algorithms that found texture patterns of canvas surfaces and the number of layers of paint.

Some have been apprehensive about the soulfulness of the recreation. And, to be honest, expecting people not to freak out at machines mapping their unrelenting beauty and chaotic shifts in behaviour, would be foolish. "I don't think we can substitute Rembrandt – Rembrandt is unique," said Mr Flores, director of technology for the project. He added, "Our goal was to make a machine that works like Rembrandt. We will understand better what makes a masterpiece, a masterpiece."

Also as a Microsoft representative rightly pointed out, "Data is used by many people today to help them be more efficient and knowledgeable about their daily work, and about the decisions they need to make. But in this project, it is also used to make life itself more beautiful. It really touches the human soul".



What's to be appreciated here, is not just the fact that deep learning has successfully recreated an artist's work, but that a team has come together to do so. *Just for the heck of it!*

KEERTHANA GANESAN
SHILPA SWETH

Alumni Interview

KRISHNA / CONSULTANT/IBM

-SACHIN ASHOK

Can you tell us about yourself? How exactly did your career at IBM start?

I am Krishna, from Haryana. I passed out in 2001 with an MCA. After that, I spent a year or so with a small start-up company at Chandigarh. After that I moved to Noida and worked with infogames for about 2 years. In 2005, I moved to Bangalore where I worked with Java on typical technical stuff. After that, I got a chance to work with IBM India in 2007. Initially I was into Java and technical stuff like web services and service oriented product development. After a couple of years, I shifted to analytics and cognition.

We would be thrilled to know more about your role in IBM and cognition.

IBM has created a huge cognitive footprint with IBM Watson. Watson is a cognitive technology that processes information more like a human, than a computer. Currently, it has a lot of cognitive start-ups under its umbrella.

So, why exactly do we need Watson? Well, there are a lot of professionals and experts like doctors, engineers, biometric engineers, finance experts, lawyers etc, right? They need to go through loads of information that has already been gathered and they waste a lot of their valuable time processing it. For instance, imagine a very busy doctor who has only 5 minutes to see a patient with a 5 page medical history. You can't expect him to read the history thoroughly, examine the patient and prescribe appropriate medicines in 5 minutes. This is where a software tool would come in handy. What Watson would do is summarize all this information and give it to you in an easy-to-comprehend, visual format, helping professionals to grasp the issue at hand very quickly. IBM has been researching on this for almost 8 years now, and that's what I've been working on.

Initially, I was working on product development and now, I'm into consultancy. I talk to clients, making them aware of how their organization can become "cognitive".

You've worked in both start-ups and full-fledged organizations. What do you think are the differences between them, in terms of experience?

Apart from the technical learning, I think start-ups give you a lot of courage to deal with obstacles. Back then, start-ups weren't something to look forward to, but the general practices have remained the same. I owe my current career to the two years I spent at the start-up. You cannot give up in a start-up. You cannot complain about not having licenses or the deadlines being too sharp. You have to get it up and running. Dot. This never-give-up attitude will take you places.

In my opinion, it's a good idea to join a start-up in the initial 2-3 years of your career, as there is very little pressure from the society, and your parents are still supportive. You are also enthusiastic about learning new technologies. When you get older and if somebody asks you to learn python when you're working in analytics with Java, you would say, "Why? What for?". [Laughter]

It is also important to think about the work culture and not just money, when you decide to settle down. For example, huge E-commerce companies generally focus on a single domain and restrict your thought process. Hence, I personally feel that it's better to get into giants like IBM, Oracle, Google etc. that provide stability only after getting some hands-on experience at start-ups.

Despite there being an increase in the number of women who've taken up engineering, the percentage is still relatively small. What do you think will engage women to actively take part and stick with it, and not change fields?

I think passionate engineers will never leave. I think it has something to do with women's psychology. I mean, say you're throwing a ball at a boy, he'll catch it. A girl will instinctively try to protect herself. That's the difference. If you're always trying to protect yourself by staying in your comfort zone and avoiding hurdles that come your way, you'll never achieve anything. We can no longer blame it on male-chauvinism as the society has come a long way from where it once was. In my career, I've worked with 4 different companies and not once have I felt any bias against women. Nowadays people actually think that women are better administrators and good executers.

#Is there anything in the past that you wish you could change? Is there something that you regret, career-wise?

Well, I regret not having studied in a nice college such as this. I was not a very studious person. I was more application oriented. Even during my MCA, I used to rarely ever listen in class and I used to bunk a lot. If I had listened more, things might have been different. [Laughter] Seriously, your college is wonderful. You guys are extremely lucky.

Any words of advice, anything you want to tell students of our college?

I would say, try new trends and don't be bookish in nature. You need to armour yourself with not just technical knowledge, but soft skills as well. Today it's not just about writing a program in Java. What is important, is to learn their connection to business. If you're trying to solve a problem, first realize how the solution will actually change the world.

It is also important to learn to communicate well. My boss used to tell me that there are two things one needs in order to be successful- Substance and Style. If you have both, you are the luckiest person in the world. Even if you don't have substance but you have style, you are prepared. Forget being nerdy and become expressive, become an extrovert. It's important for the industry.

#Do you think the curriculum has to change to bridge this gap between institute and industry?

I don't think so. You may think - Where are we going to use all this theory? We're not going to design another UNIX, right?

The important point is that you are able to understand the connections. Today when I study machine learning or write an algorithm, I can relate it to what I've learnt in discrete structures. The actual problem is that majority of you bind yourselves in a well. You don't look outside. In your holidays, go to cities like Bangalore, Noida, or Gurgaon. Meet people, industrialists, engineers. It will give you an insight to how the world actually works. Browse forums, participate in competitions, hackathons etc. The aim is not to bring home goodies, but rather, knowledge. Keep your eyes and ears wide open. That's how the gap can be bridged.

**- Anugrahaa
- Lakshmanaram**

AWKWARD GREGORY

and TRANSACTIONAL MEMORY

-AYUSH MISHRA



Gregory is a nice man. He minds his own business, is always ready to help others and always means to be polite. However, he is often caught in awkward situations.

One of them, is a situation frequently encountered by you and me, though much more frequently for our awkward and indecisive Gregory. Every now and then, he has to visit the photocopier in the office, for which he must pass through a narrow hallway – a place that has, but isn't designed to support two-traffic. And this is where the problem is (though very small, but very big in awkward Gregory's head).

Say Gregory enters the hallway from one end while his boss is returning from the photocopy room from the other end. They start moving towards each other, each eager to get back to the errand they have to run. However, their swift movement is impeded, rather abruptly, when they run into each other.

Gregory moves to his right to let his boss pass, and his boss moves to the left – still blocking each other. The Boss sees that Gregory has moved, and moves back to where he was, but so does Gregory, blocking his boss once again. And again, and again.

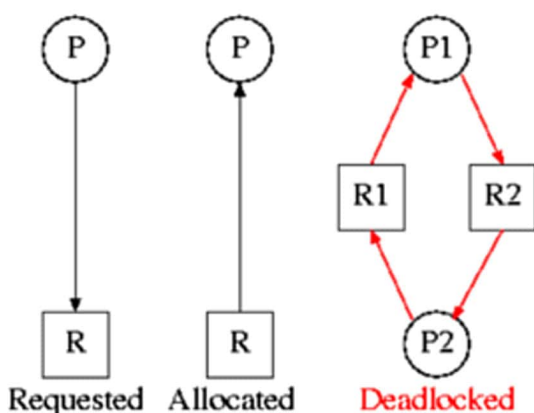
(You get what I mean)



Such Problems are often faced by Computer Scientists in Cloud Computing systems too. Cloud computing uses what is called 'Concurrent Programming', something quite opposite to the parallel computing paradigm. On one hand, where Parallel computing aims to process a thread parallel in multiple processors to save time, 'Concurrent Programming' processes multiple threads in a single processor.

It has to do this because in cloud computing architectures, the multiple processors on all the multiple machines are shared between the numerous processes running on the cloud. Think of it this way - in a restaurant, a chef has numerous assistants to assist him in each subtask in making the meal (chopping vegetables, boiling, garnishing etc.) and these assistants work in parallel to save time (Parallel Cooking). On the other hand, when your mum simultaneously cooks the dal, boils the rice and sauté's the veggies, she's concurrently cooking.

Say there are two Threads (or processes) T1 and T2 that must access the same memory location in the database and write to it. This is something like both these threads trying to pass through the same narrow "Cloud Hallway" where each thread can't successfully complete its task because of interference from the other.



This problem is solved by introducing Atomicity in these threads. What this means is that, both the threads should run in a way that it appears indivisible and instantaneous to the other thread. Execution of these atomic threads is called a 'Transaction'.

Luckily enough, Gregory happens to be a cloud computer Scientist, and assesses his situation like two 'Dead Locked' threads. This is what he notices:

Gregory and his Boss approach each other

Gregory sees that the right hand side of the hall is empty

The Boss sees that the Left hand side of the hall is empty

Gregory moves to his right

The Boss moves to his left

They block each other again

He notices that his and his boss's thinking is intertwined like two threads scheduled to be processed on the same processor. Applying the principle of atomicity, he comes up with a more fool proof algorithm to overcome this situation.

**Gregory and his
Boss approach each other**

**Gregory sees the right hand
side of the hall is empty and
moves to the right**

**Gregory's boss sees that HIS
right-hand side of the hall is
also**

They both pass

And this is how Cloud Computers solve conflicts through Transactional memory (TM) too! Sometimes such operations lead to wrong values being read – which is easily taken care of by creating a log that stores meta-data of the 'version' of each 'optimistic' read (read without checking) and 'pessimistic' write (write with possible roll backs).



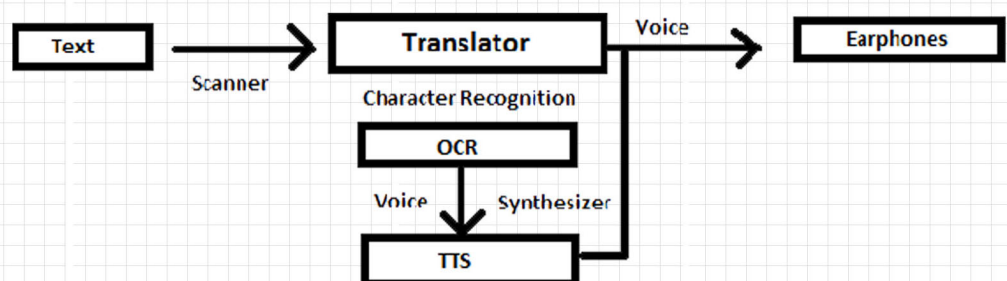
**ADITYA BALAJI
SHILPA SWETH**

VOICE STICK

-By: Rahul Kar

Rahul is deep in thought in his cabin about what to gift his beautiful sister Shreya, on her 18th birthday. She loved books, especially fiction. But they all had to be in Braille as Shreya has been denied the gift of vision since birth. She had already read most of the affordable Braille books and Rahul knew his gift had to be something related to this. Just then, Ayush, the tech-buff and his best friend, bursts in through the door and exclaims, "The Voice Stick! That's what you should give her!" Well, we're not sure if Rahul understands what a voice stick does, so let's give him a tour of the features of the voice stick.

The Voice Stick is a concept text scanning device for the visually-impaired. When the stick scans in printed letters, the OCR (Optical Character Recognition) function recognizes the text and converts that information into voice.



Voice Stick

Fig:-How it Works

HOW IT WORKS?

It basically functions in three steps:

.TEXT-INPUT:

First, the stick is placed on the content to be read. The scanner scans the text and sends the data to the translator.

.TRANSLATION:

First, the Optical Character Recognition (OCR) technology scans typewritten text and uses mechanical or electronic methods to translate it into machine-editable text.

The Voice Stick then uses text-to-speech (TTS) technology to translate the scanned written text into a voice spoken rendition.

.AUDIO-OUTPUT:

The voice obtained through this translation process is read back through a pair of earphones which come along with the stick.

WHY?

1. Cost: The Voice stick is more cost-efficient than Braille books.

2. Portability: The Voice Stick is just the size of a normal pen, and can be carried easily.

3. Variety in application: The Voice Stick is not only useful for reading books, it can also be used for e-mails, newspapers, contracts, etc.

So there's that! What could be a more precious gift for our beautiful Shreya? On receiving the Voice Stick as a birthday gift from her dear brother, Shreya was ecstatic. She felt a tremendous amount of joy as she was now relieved from all the Braille reading, and was reminded of those days when her mother used to read to her.

RECENT EVENTS



- Arvindmani

29th January saw the 2016 edition of the Computer Science and Engineering Department's annual symposium, Vortex, in its largest display yet. Vortex '16 seemingly outdid the previous editions in almost all aspects, encompassing several guest lectures, competitions, paper presentations, and workshops.

Its growth can be followed through the increase in the impact it has had online. The Facebook handle for the symposium ballooned in its activity and reach, helping Vortex become what it did.

The solid guest lecture line up included many of the biggest names in various fields of computer science and technology, including but not limited to, the Director of Amazon in India, and the founder of DoSelect.

A special initiative this year was the Career Guidance Counselling by seniors of the department for the students only just entering the vital stages of their education. Another notable first was the Social Responsibility Initiative by Vortex to aid those in need during the harsh times of the Chennai Floods.

When asked for the highlight of this year's symposium, the Chairman of CSEA Mr. R. Santhanagopalan replied unhesitatingly, 'Definitely the internship opportunities. We provided internship opportunities for winners of every single event in Vortex. Amazon provided direct internship for the winners of the OSPC.'

On the growth of Vortex, he added, 'I've watched the symposium grow firsthand, and I've seen the work put in by the team to nurture this growth. The Treasurer, Harish, can attest to this. It has given me a sense of pride, accomplishment, and personal satisfaction for having worked for Vortex during my years of study in this college.'

ADITYA BALAJI
MANJITH DUNGUNG

Universal hashing

Parth Thakkar

Hashing as you know, is a very useful technique of storing data. It is something that lets programmers to both have their cakes and eat them too – small space requirements and also fast access times. In this article, we are going to look at a method of hashing that makes it even more useful – universal hashing.

To be on the same page, a quick review of hashing – it is a storage technique where a set of ‘keys’ (data) is stored in a hash table. Hash table is an array of linked-lists¹. Let’s say you have n keys, and the size of the hash table is m . To store the data, you define a hash function $h(k)$ which takes a key and maps it to some number in the set $\{0, 1, \dots, m-1\}$. So, you store the key k in the linked list located at index $h(k)$ in the hash table. Insertion is thus $O(1)$ – you calculate $h(k)$ and then insert the key at the beginning of the linked list. Lookup time depends upon the number of ‘collisions’ – the number of keys that are mapped to the same location. In the worst case – where all keys get mapped to the same location – lookup takes $O(N)$ operations. In the best case, it is $O(1)$.

Hashing is all nice and good, but the problem is choosing a good hash function. If you choose a bad hash function, then the access time is going to be slow. In fact, there is no single hash function that will be good for all sorts of inputs. Even worse, if there is a single hash function, then one can always choose keys that are mapped to the same number by the hash function – giving an $O(N)$ access time. Universal hashing overcomes that by choosing a hash function at random from a family of hash functions! If there isn’t a fixed hash function, it isn’t possible to choose keys that are mapped to the same number by the hash function!

Consider this example: You have to perform hashing on 250 IP addresses. Let’s choose the size of the hash table to be $n=257$ – notice that it is a prime number. To define our hash function, we will think of an IP address as a tuple of four numbers $X = (x_1, x_2, x_3, x_4)$, where each $x(i)$ is less than 255. So, the IP address 128.109.29.68 should be thought of as (128, 109, 29, 68). The hash function is then defined as

$$h_a(X) = (a_1 * x_1 + a_2 * x_2 + a_3 * x_3 + a_4 * x_4) \pmod n$$

Each $a(i)$ is a number less than n . This then defines a family of hash functions H . It turns out that if you pick a hash function randomly from this family, it is very likely going to be a good hash function! That is, the probability of $h_a(X) = h_a(Y)$ where X and Y are different is $1/n$ if h_a is randomly chosen from H .

¹I’ve described the chaining method of collision handling in hashing over here. Of course there are other methods of handling collisions, but the idea here is to reduce the number of collisions, and not how they are going to be handled.

The proof of this is simple:

If X and Y are different, they differ in at least one component. Without loss of generality, let's say $x_4 \neq y_4$. Then, for $h_a(X)$ to be equal to $h_a(Y)$:

$$\sum_{i=1}^4 a_i x_i \equiv \sum_{i=1}^4 a_i y_i \pmod{n}$$

That is,

$$\sum_{i=1}^3 a_i (x_i - y_i) \equiv a_4 (x_4 - y_4) \pmod{n}$$

Now, let the LHS be of some value c. Now, since n is prime, $(x_4 - y_4)$ has a unique inverse modulo n. Hence, $a_4 = c * (x_4 - y_4)^{-1}$. The probability of this happening is clearly $1/n$, and hence we're done!

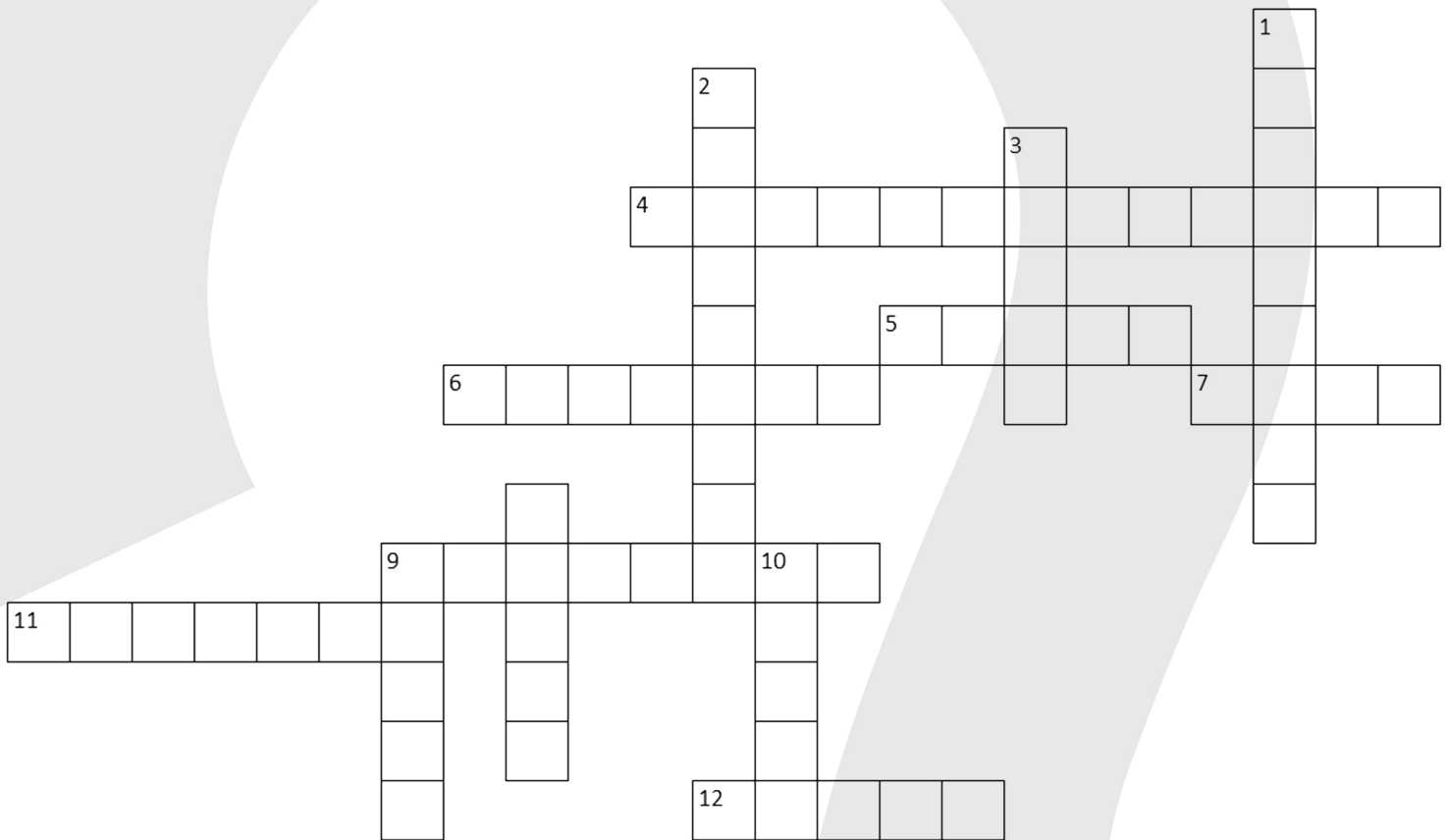
What we saw here was the scheme of Universal Hashing. The set H we defined above was an example of a Universal Hashing family of functions. A set of hash functions H is said to be universal if for any two data items x and y, exactly $|H|/n$ of all hash functions in H map x and y to the same bucket, where n is the number of buckets.

There is another interesting type of hashing called Perfect hashing, which gives $O(1)$ performance not just in expectation, but even in the worst case! Of course, it has limitation – it works best for static keys. We won't be touching up on that in this article though. So, for now, that's all folks!

Editor : Thivyavignesh
Designer : Pradhiksha

Its all about the “algos”

Answers are names of algoritms (without spaces)



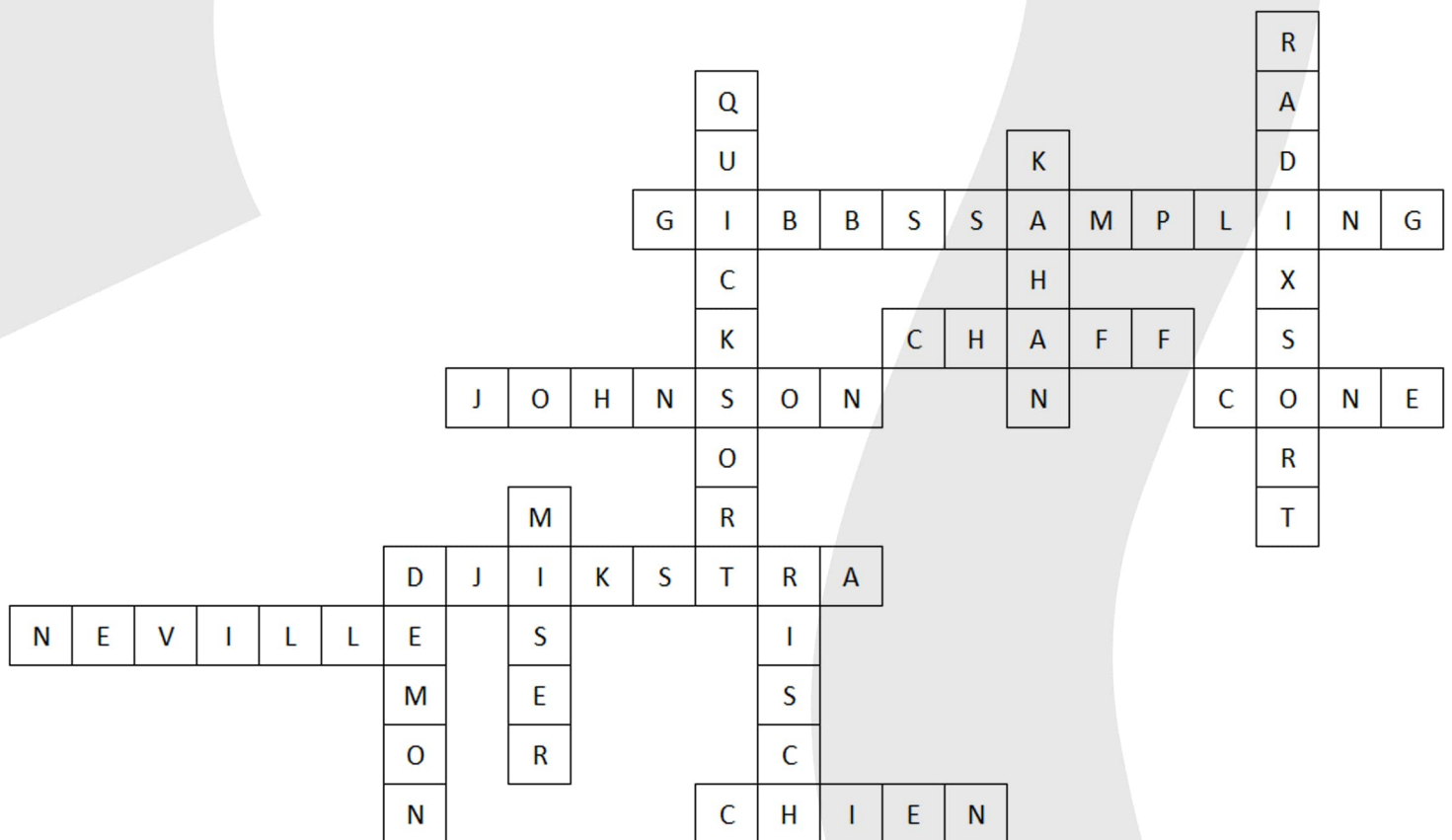
ACROSS

- 4** Generate a sequence of samples from the joint probability distribution of two or more random variables
- 5** An algorithm for solving instances of the Boolean satisfiability problem
- 6** All pairs shortest path algorithm in sparse weighted directed graph
- 7** Identify surface points
- 9** Computes shortest paths in a graph with non-negative edge weights
- 11** Polynomial interpolation algorithm
- 12** A recursive algorithm for determining roots of polynomial defined over a finite field

DOWN

- 1** Sorts strings letter by letter
- 2** Divide list into two, with all items on the first list coming before all items on the second list, then sort the two list
- 3** Hindi word for “where”
- 8** A person who hoards wealth and spends as little money as possible
- 9** Other name of devil (starting with d)
- 10** An algorithm to find indefinite integration

ANSWER



**Bits & Bytes now welcomes
freelancers to send in their work to be
included in the subsequent editions.**

Contact Us:

www.facebook.com/csenewsletternitt

csenewsletter@nitt.edu

