threads.

threads.nitc.ac.in

SEP.

2018

PREFACE

We are the 'Digital Generation'; the one that can access information with the swipe of a finger, the one whose thoughts and actions are marked online, and whose daily routine is woven together by technology. From the ad suggestions popping up before our screens, to the data analytics used to win presidential elections, computer science proves to have a hand in every facet of our lives. As innovations keep sprouting from across the globe and technology advances, we remain oblivious to a vast portion of its capabilities.

Threads, the official newsletter for Computer Science and Engineering Department, thus weaves together intriguing tidbits from the tech world and problems to wrack your brains. At the same time, our department at NIT, Calicut remains to be a powerhouse of activities, which we list proudly in the Department Buzz section.

This edition takes a peek into the ubiquity of Machine Learning in today's world, the controversy of Cambridge Analytics, and the deeper possibilities of Git Reset for the git enthusiasts. While the Art Gallery problem, by Dr. Subhasree M, may pique the interest of the lovers of problem-solving, the CSE Crossword tests your wit and could get you an exciting prize. We also have an article by our alumnus Ms. Anjana Soman, where she takes us through her last day as a student here at NIT, Calicut. Lastly, for the competitive coders who can’t get enough, here’s a Coding Calendar to gear up for the upcoming challenges.

While you browse through the newsletter, we hope you too, will join us in the journey of learning and exploring the ever-expanding world of computer science.

Suggestions and submissions for the next edition are welcome at csea@nitc.ac.in

• • •

DEPARTMENTBUZZ

SEPTEMBER

An MOU was signed with Sree Chitra Tirunal Institute for Medical Sciences and Technology (SCTIMST) on the 11th of September, 2018, in healthcare and allied fields. Following the same, a talk by Dr. Harikrishnan, Professor of Cardiology, SCTIMST, on 'Introduction to Heart Diseases and Challenges in Cardiac Intervention' was held by CSEA. View the full report at http://nitc.ac.in/?url=news/view/1903/1

AUGUST

Dr. Saleena N took over the position of **Head of the Department** for Computer Science and Engineering Department in August 2018.

Mr. Shiju Kakkadath, a batch '99 student of REC and presently the Lead Consultant of Wipro Limited (Zurich, Switzerland), gave a talk on **General Data Protection Regulation** and its implications to Banking Industry on 13/08/18. The event was hosted in association with IEEE Malabar Chapter.

The 4th of August marked the **Silver Jubilee Reunion** for the CSE batch of 1993, Relive '93. Around 20 alumni presently holding high positions in the industry gathered together on this day for a walk down the memory lane.

JULY

24th July 2018 saw the **Inauguration of CSEA Activities** for the academic year 2018-19. The ceremony was followed by a keynote address by **Mr. Venugopalan U**, an alumnus of REC Calicut (1992), on **Virtual Extensible Local Area Network**.

DEPARTMENTBUZZ

CSED has recruited six new faculty members this year. The recruitment process has taken place after a long span, and the department extends a warm welcome to the new members.

Arun Raj Kumar P
Assistant Professor
Ph.D (NITT)
https://sites.google.com/site/park286/

Hiran V Nath
Assistant Professor
Ph.D (IDRBT)
http://people.cse.nitc.ac.in/hiranvnath/

M Prabhu
Assistant Professor
Ph.D (Anna Uni.)
http://people.cse.nitc.ac.in/prabhum

Vasudevan A. R.
Assistant Professor
Ph.D (NITT)
http://people.cse.nitc.ac.in/vasudevanar/

T Veni
Assistant Professor
Ph.D (NITT)
http://people.cse.nitc.ac.in/veni/

Jay Prakash
Assistant Professor
Ph.D (ABV IIITM)
http://people.cse.nitc.ac.in/jayprakash/

JUNE

CSED, in connection with IEEE Power and Energy Society NITC Student Chapter organized a talk on **Crisis of Values** in the 21st Century by **Dr. V.P. Joy IAS**, Central Provident Fund Commissioner from Employees Provident Fund Organization (EPFO) on the 29th of June 2018.

A Faculty Development Programme on **Security and Privacy in Big Data Analytics** was held during 18-23 June 2018, sponsored by Technical Education Quality Improvement Programme (Phase III). The training was also twinned with Department of CSE & IT at Govt. Engineering College Bharatpur.

Recursive Wayfarer

Anjana Soman
B-Tech, Batch of 2018, CSE

It's only fair that the sun beat down mercilessly on such an occasion. Its best intentions cloaked in mellow warmth that adds the slowburn to the unfolding drama, it watches the larger share of a roughly 1000-strong populace scuttle about getting their papers in order. For after all, dear Anne, outside the annex, identification papers help you toe the fine line between freedom and enslavement.

As I stand in line outside the tiny little photostat center crammed with all my fellow scuttlers grabbing their own plethora of empty forms, it isn't thoughts of the aftermath that flood my brain - the day grants very little screen-time for any thought not directly pertaining to it. Instead, I'm trying my best to convince myself this isn't some shoddy reenactment of Zack Snyder's Suckerpunch, that we aren't all making a mad dash out Lennox House, searching all over campus for the secret items that make up a plan of escape. The procedure set in place for the application of the provisional degree certificate doesn't fall too short of being an adventurous quest itself, sending the students across campus in their bids to collect the coveted No-Dues stamp on all their numerous forms.

But distant bells ring in my head, signalling the beginning of the ritualistic circumambulation of the campus that now awaits me, for I finally have an empty set of forms in my hand. I'd done my waiting. 4 years of it. In NITC.

'Twas now time.

I ghost through the first few stops, trying to recall what exactly constituted our PE sessions back in the good ol' days. First year in its entirety often feels like a haze, an extended ritual designed to break in the fresh set of greenies. Squint hard enough at the memories, and a familiar set of faces that grew to become one's college clique looms into view; fond smiles and cringe-worthy mishaps. What really sets the tone for the college experience though, is second year. Second year when you finally, finally get to learn what you signed up for and revel (surely, you're CSE?) in your choice of department. But as I make my way to the Library, praying I've returned the few books I chanced upon in there, I can't help but recall how wet behind the ears we were when we made the leap to second year.

Never let it be said that college doesn't prepare you for the real world, for the race to the library for the meagre copies of the prescribed reference books for each course is as cut-throat as it gets. It's the subtler lessons that ingrain themselves into your psyche. Till date, I get library-sprint flashbacks whenever a time-bound service is announced.

Trudging through the numerous labs in the IT complex that have amassed the envy of the rest of the campus gives me an opportunity to reminisce. Sure, I'm running to get the elusive stamps and leg it out of the place, but spending whole semesters holed up in those rooms gives them an intangible hold over you. I take a few moments to greet the lab staff who've often played Saviour to our countless code crises, and take a few last whiffs of the air-conditioning now that the labs are as

empty as I'll ever find them. It's with slight amusement that I head back to the predecessor, the Central Computer Centre, the holy grail of Internet access back in first year. I wave my forms defensively when the security guard raises an eyebrow at my unlogged entrance, and he's instantly assuaged. What'd I tell you, Anne? Papers, it's all about the papers.

A profound lesson college imparted to me was that pain is inevitable.

It's a visceral, gut-wrenching ache to write down my NITC email ID in the CNC register, knowing they'd be deactivating it by the end of the day. And as much as I'd miss the frequent emails about thesis presentations and the Lost and Found, it's triumphed by the greater pain of losing access rights to the Campus Wifi. And just like the world, the college doesn't tolerate freeloaders, kids. There's no such thing as free Wifi. And if there is, you're better off not using it.

But if losing Wifi login credentials was pain, then the Hostel procedures are torture. Not so much the constant running back and forth between the Main Hostel office and the multiple LH offices, but the books. Oh Lord, the stacks and stacks of books. Nay, ledgers. Ledgers bleeding ink, when sleek computational methods sit right within our reach. The small room inside the Main Hostel office where students go to plead their case regarding their mess bills is split into two halves - humans and ledgers. Would it be hyperbole to say it takes me longer here than the rest of the whole quest combined? Yes. Would it stop me from saying it? No.

There are many names to take, so I shan't take any, but I talk to them. Promises are made to visit now and then, wishes sent forth, and just like that, I'm all done.

Well, almost.

I come full circle, and head to the Academics department, stamped set of forms in hand. My hand is spotted in ink, but it isn't shaking when I hand over the ordered, stapled document set to the harried soul at the desk. My bag is lighter when I step out into the evening sun, but my heart is just a tad heavier. A whole day of scurrying about later, I suspect I'll convulse if I were to hear even the slightest ruffle of papers in the near future. But I take a good long look around, as one ought to at the end of such an occasion, and wonder what it's going to be like.

Months later, when I stride past the main gates, it looks like everything's changed. The world is transient after all. But ten strides in, twenty strides in, and it's like I've never left at all.

• • •

Why Machine Learning?

Mohammed Fayaz Salim
B-Tech, 3rd Year, CSE

The advent of the computer has been a boon to mankind in many ways. We have been able to predict weather patterns, build 1000m towers, mould and design nature as we please and so on...these feats would have been unthinkable of in the past. Certainly, at the heart of all these achievements lies the computer – granting us unthinkable computing power and granting us access to feats of engineering that would have been unfathomable just a century ago. At the heart of the computer lies the algorithm – algorithms have been designed in recent times that have made things that most of us take for granted that would not have been possible otherwise. Take for example video streaming, not many of us know that it is made possible by a marvelous algorithm developed in the 20th century whose roots can be traced back hundreds of years – The FFT (Fast Fourier Transform).

However, in spite of all our ingenuity, some problems seemed too difficult to solve for even the most skilled programmer. After all, algorithms were simply hard coded rules and hence given a scenario with millions of outcomes it would be impossible to account for the sheer diversity of possibilities. Consider the following simple problem – Given 2 images of an apple and an orange respectively, classify them. This seemingly simple problem was unthinkably difficult to solve via classical programming. Let us take a crack at the problem anyway just to see what the issue might be. Let us try to classify them on the basis of their color. We can keep track of the number of pixels of each color (which would be red, green or blue for a RGB image) and based on the proportion of pixels, we could try to classify them. OK...this would work for a perfect sample – one where there is a perfectly orange orange and a perfectly red apple. Furthermore, these fruits would have to be the only objects in the image. This is heavily limited and would serve no purpose in the real world – The real world is messy and there's a lot of variation. Apples can come in different colors, there may be other objects in the image, the images may be black and white just to name a few scenarios that would break our admittedly simple algorithm.

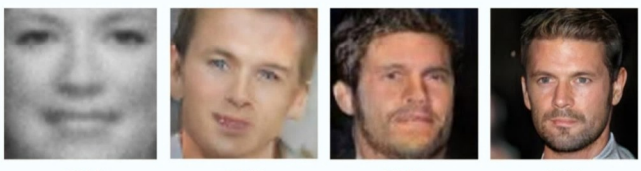
Thus, begins the search for a new programming paradigm that would be able to solve these problems in reasonable time and with reasonable effort. This search is what gave birth to the programming paradigm that we now call Machine Learning. How do we solve the issues that plagued classical programming in image classification? Simple – We learn by example. We feed the algorithm examples of what an apple looks like or what an orange looks like and the program iteratively learns that the image contains an apple or an orange. Granted that in order to gain a sufficiently high accuracy of classification we would need to feed in a large number of images of apples and oranges, but this in practice turns out to be far more feasible and accurate than hard coding rules for classification.

Machine Learning has turned out to be immensely useful in problems not relating to image classification too. Ever wondered how ads are tuned to your preferences? Machine Learning. Ever wondered how YouTube figures out how to recommend videos? Machine Learning. In fact, Machine Learning has become ubiquitous in today's world – being used in tandem with a host of other algorithms to make our lives simpler.

Let's now take a deeper look at how exactly Machine Learning works. On a high level, machine learning algorithms can be classified into 2 – Supervised and Unsupervised algorithms. Put simply, in supervised learning we know what the data are, and we do not know what the data are in unsupervised learning. We will take a look at one very simple Machine Learning Algorithm – The Decision Tree Classifier. Let us take the previous example of classifying a fruit as an apple or an orange. Let us assume that apples and oranges have two important properties – texture and size. Now let's assume that apples have a smooth texture when they have a size less than 5 cm but they are bumpy otherwise. Let's assume that for oranges, they have a smooth texture when they are below 5 cm but they are otherwise bumpy. Now the problem of classification can be solved by a decision tree. Simply ask a question about the properties and branch as necessary, narrowing our possibility space on each decision. That's all there is! Granted that there are more sophisticated algorithms that perform more complicated kinds of classification, but this is, in essence, all there is to machine learning.

The decision tree classifier that we have just seen above is inadequate for most applications in the world and hence we require more powerful methods of learning that can learn even more complicated relationships. Hence, in the real world we turn to neural networks and deep learning to solve more complicated problems such as computer vision and image recognition.

Now, we will look at another recently found use for machine learning. Machine Learning until very recently was used primarily (almost exclusively) in classification problems. But say we want an algorithm to produce novel data (i.e. data that the algorithm has never seen before) - Enter Generative Adversarial Networks or GANs for short. GANs allow the creation of novel data from examples. The working of GANs is quite interesting so we shall take a quick look into how GANs function. GANs have two neural networks working in a zero-sum game framework. One network is called the generator while the other is called the discriminator. The discriminator is trained on the examples that we provide and then we set up the generator to produce data. The goal of the generator is to produce novel data that appear to have come from the distribution of data that we provided to the discriminator. At the end of this process, we will end up with a generator that can produce data that appear to have come from the distribution that was provided to the discriminator before this process.



GANs have been quite successful in producing images of people that have never existed and landscapes that don't exist in real life just to name a few of its successes.

You should now have an understanding of why Machine Learning is ubiquitous in today's world and how it's shaping our industries and our lives as we speak. I leave you with this quote from Nick Bostrom "Machine Intelligence is the last invention that humanity will ever need to make."

Working with Git: Reset Demystified

Abhiram Haridas
B-Tech, 4th Year, CSE

Git: it's the most widely used version control system out there. This article is intended for users of git, who might be interested to know how *git reset* actually works. In the process we'll take a sneak peek into basic git internals. The reader is assumed to have a basic understanding of the terminology and workflow associated with git.

With that being said, let's get started. The basic git workflow goes as follows, initially you make changes to files and save them. Then comes the staging area. Staging area is the *proposed next commit*. This is how you tell git that these are the changes (or files) that you wish to see as part of the next commit. Then when you commit, you essentially sync the commit history to be up to date with the staging area. Internally git maintains three trees ('trees' as in a hierarchical collection of files not the data structure) to handle this workflow. The working directory tree as the name suggests keeps track of the current state of the working directory. The index tree keeps track of the staging area. The tree that keeps track of the commit history is called HEAD. Technically HEAD is a pointer to the tip of the current branch (in fact it can point to any arbitrary commit, more on that coming up) which tells git where to attach the next commit to.be used interchangeably and same is the case with words 'commit history' and 'HEAD'.

In git terminology and in this article, thet words 'staging iarea' and 'index' will be used interchangeably and same is the case with words 'commit history' and 'HEAD'. When a commit is made all three trees end up in sync.

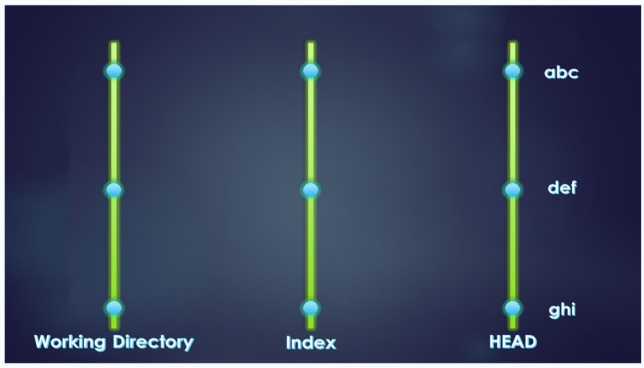


Figure 1

Now we are in a position to take on *git reset*, which is one of the most widely used yet poorly understood git commands ever. Firstly there are three modes of reset, soft reset, mixed reset (which is the default one) and hard reset. The functionalities happen in an incremental manner. The mixed reset does everything soft reset does and tries to do something more. The hard reset does everything mixed reset does and does even more.

Consider the situation illustrated in Figure 1. Currently all the three trees are in sync. The strings 'abc', 'def' and 'ghi' represent three different commit Ids. *git reset –soft def* (See Figure 2a) would modify the HEAD tree to end at the specified commit (i.e 'def'), leaving the working directory and the staging area untouched. In this case since we are resetting to the second last commit, we are essentially undoing the last git commit command.

This can be used as an alternative to the widely popular *git commit –amend* command which is used to edit the very last commit. After the reset, we could stage more changes and commit again with a different message.

single new commit with the changes combined. If you wish to modify the files staged in each commit, you could use mixed reset which would undo the earlier staging and then stage the files as required again.

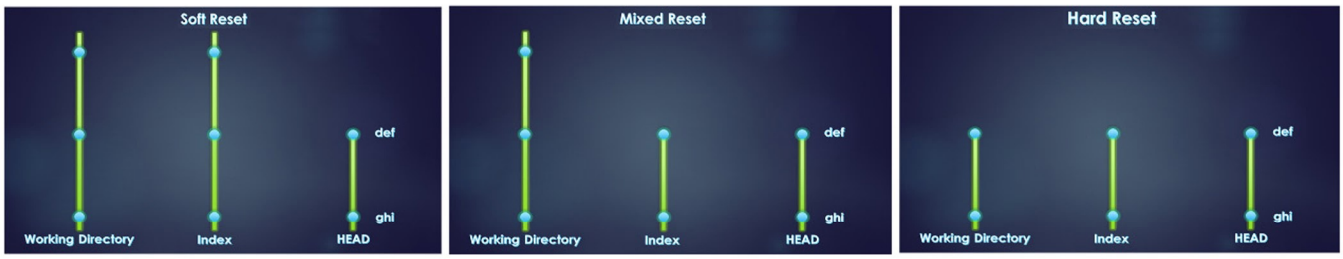


Fig a Fig b Fig c

Figure 2

git reset –mixed def (see Figure 2b) does something further. It also modifies the index tree so that the staging area also ends at the specified commit. Again in this case (of resetting to second last commit) this would have the effect of undoing the last *git commit* and *git add* operations. The working directory is untouched. Now you can probably guess what *git reset –hard def* (see Figure 2c) does, after doing what the mixed reset does, it proceeds to change the working tree to end at the specified commit. Now all the changes since that specific commit have been reverted. And that's pretty much it about how reset works.

git reset –hard (which defaults to *git reset –hard HEAD*) is widely used to revert all changes in the working directory since last commit. There might be several other use cases of reset depending on what you are trying to achieve but we put an end to our discussion of use cases here.

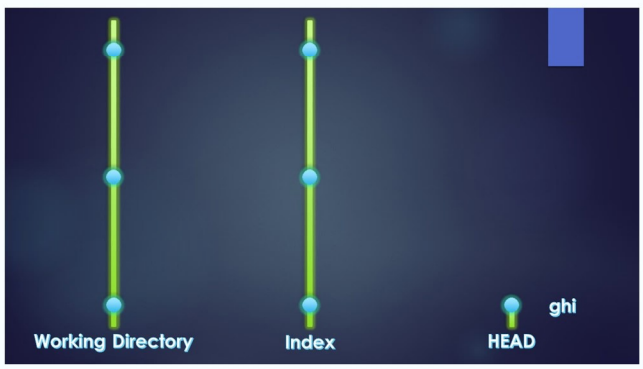


Figure 3

Having this level of understanding about reset enables you to do a lot of things with it. As we have already seen, we can essentially uncommit or unstage specific changes. Consider another scenario in which you wish to merge two recent commits, reset again holds the key. In the above example, *git reset –soft ghi* (see Figure 3) will undo the last two commits, with the working directory and staging area untouched. A *git commit* command would now create a

A couple of important things to note here: one is about the usage of git to modify history like the case in which we combined two commits into a single one. New commits and being created in place of the old ones. This is not encouraged if the previous commits are already pushed to a remote. Modifying history which has already been

published online is a bad idea since there might be other people basing their work off it and it creates issues if you modify them. Even in the case where we undid just the last commit, committing again would give you a new commit resulting in a divergent history from the last one. The second important thing is that there are very few things in git that cannot be undone and hard reset is one of them. Usage of reset can be dangerous and you might lose your work. Technically those commits are not lost i.e you can go back to them even after a reset given that you know the commit ID. For that you probably need to have superhuman memory to remember the hash values or you might be lucky enough to have a log output listing them in your terminal window, both of which are unlikely.

There is also an interesting similarity between git checkout and the hard reset. Both of them seem to take the three trees to a specific commit. Before we try to distinguish them, let's take a closer look at checkout. Branches in git are just pointers to specific commits. HEAD is also a pointer which usually points to one branch pointer. When you checkout a branch, the HEAD pointer is moved to the specified branch pointer and the working tree and index is also moved to that commit to which branch pointer points to. You can also checkout specific commits rather than branches. This puts the repository in the infamous detached HEAD state meaning that the HEAD is not pointing to any branch pointer (This is undesirable because the commits made from this state are not associated with any branch and become unreachable once you checkout any other branch or commit.). Now coming back to the similarity with reset, we find that in both its use cases (checking out a branch or a commit), checkout does not affect the branch pointer that HEAD was currently pointing to. For example if you checkout 'newbranch' from your 'master' branch, then the master branch pointer is not affected in any way. Whereas in reset, if you reset to a commit from master, then the master branch pointer will now point to that specific commit which makes the commits following that specific commit unreachable.

References

This article is based on discussions about reset in **Pro Git by Scott Chacon and Ben Straub Second Edition Apress, 2014.**

Illustrations by Anupam Asok

• • •

The Digital Age Coup d'ètat

Sreelal E S
M-Tech, 3rd Year, CSE

Wes Ball is a resident of Ohio, USA and has been my Facebook friend for over two years. During the last presidential election, out of pure curiosity, I asked for his political views and predictions on the election. Much like many of our youth, he wasn't much concerned about the results, not because he didn't have any concern about the future of his country but he was too realistic. Whether it was Trump or Clinton, United States was about to take a step down from Obama's policies, at least that's what he hinted. The results were a close call and Trump became the president. That day passed on as any day would, except with some rumors over Russia's involvement in the electoral process and I thought, here we go again, after 'Fake Moon Landing' and 'JFK', we have yet another conspiracy in our hands. Then, after 12 months, Cambridge Analytica happened.

The Sting

What started off as an undercover scoop by Channel 4, turned out to be one of the most shocking news stories of this millennium. CA, with its political and financial analytical wings, has already drawn a lot of attention in 2017 itself, but their tie-up with Facebook and how they have used user information for their seemingly nefarious practices was the core issue.

For a keen tech enthusiast, what CA did was a groundbreaking application of data analytics. They went on to analyze user preferences and emotions and found out the issues that will be the most striking ones on an election campaign. The next step was something that every businessman would approve: CA went on to supply the information to the highest bidder. Although it is not proven conclusively, this points fingers to the Russian ties in the last US Presidential elections. For where it went wrong was that CA seemingly ignored the very thing that binds a Democracy: Its people. Personal choices and preferences of common men became subjects of a bidding war, violating not just the right to privacy but the very ethics of a transparent election process. Obviously, CA denied the allegations and so did Facebook. A formal investigation triggered by the US Senate is still undergoing and where the lines will be drawn, that we will have to wait and see.

The Whistle

Whistleblowing is a courageous act and it gives the person a heroic stature considering who the nemesis are: greedy corporates, authoritarian governments and so on. From Jeffrey Wigand to Ed Snowden, it's quite discomforting to watch the list grow in the span of a few decades. Christopher Wylie is the new addition to that list, for which he was the tide that broke the wall for Cambridge Analytica. 'The Data War Whistleblower', a term coined for Wylie by the Guardian, provided crucial documentation that supported the theory that Facebook was indeed

harvesting user information for CA. How many countries came under the influence of CA, is still under speculation but according to Wylie, US Presidential elections and Brexit votes were the major ones.

The Indian Currents

While all these ruckuses were going on in Britain, the Indian populace was still pondering over the outcome of Demonetisation, stooping GDP and who's going to win the next election. Little did we knew that CA had, in fact, tested their weapon of political destruction in India. At that time, there was no CA, but the players were all the same nevertheless. They were keen to grab an offer from Congress so that they can spend their expertise in political behavioral studies in our country and at the same time they got offers to defeat Congress from an unknown source. For some unknown reasons, it never took off, but blood was spilled in this affair, some say. We still don't know the truth as our government doesn't share the enthusiasm of US Senate, which carries out a rigorous investigation on the matter.

The Tail End

Cyber Warfare is not a freshly coined-up term, as the possibility of remotely accessing public infrastructures to handicap the enemy is already tested in many battlefields. But toppling and installing governments with illegally collected private information was a new incident and a sane person can only view this as the tip of the iceberg.

Every software product we use has a user agreement and we fail to grasp the content of it every time. Cyber attacks are already taking place by capitalizing this loophole. Now, even an ironclad agreement of trust is broken for immoral purposes. Professional ethics and engineering integrity are thrown out of the window for the highest bidder of information. Information has always been an integral part on a war-front and that's why the British broke Enigma. The weapons are the ones that change; war and destruction still prevails...

Reference

https://www.cnbc.com/2018/03/21/facebook-cambridge-analytica-scandaleverything-you-need-to-know.html

https://www.theguardian.com/uk-news/2018/apr/07/christopher-wylie-whyi-broke-the-facebook-data-story-and-what-should-happen-now

https://theprint.in/politics/exclusive-inside-story-cambridge-analyticaactually-india/44012/

Art gallery problem: Variants and relatives

Subhasree M
Assistant Professor,CSED,NITC

Suppose you have an art gallery with precious paintings and sculptures. It is necessary to protect the gallery by security guards or security cameras. The specific requirement in this case is to use enough and minimum number of guards / cameras so that any one of them (between them) can oversee the whole gallery. Given an art gallery, how many minimum guards do we need to safe-guard the whole gallery?

Figure 1 : A simple art gallery : needs only a minimum of one guard (Image courtesy: Njuz.net)

It is obvious that the art gallery in Figure 1 can be safe guarded by a minimum of one guard. What about the art galleries in Figures 2 and 3?

Figure 2 : Another not so simple art gallery : needs at least three guards

Figure 3: The Guggenheim Museum in Bilbao: hard to supervise (Image courtesy: BBC.com)

Art gallery problem is a challenging and still active problem in Computational Geometry, even though it was first posed in 1973 by Victor Klee. In general, buildings are likely to be of rectangular shape, but art galleries (as shown in Figures 2 & 3) could be of diverse and crazy shapes.

To make the problem simpler, the floor plan of the art gallery is assumed to be a simple polygon (a polygon with non-intersecting edges and without holes), as shown in Figure 4. Classical art gallery problem assumes stationary guards on vertices of the polygon (known as vertex guards) who can turn around (ie. 360° visibility) and who is not able to see through the edges of the polygon (ie. see through the walls of the art gallery).

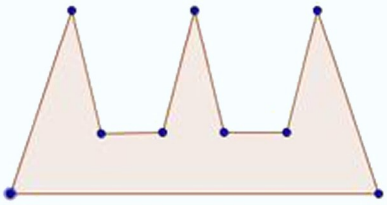


Figure 4: An art Gallery represented as a simple polygon

There are many variants for the art gallery problem: (i) Point guards: Guards can be static and be placed inside the art gallery as well as on the vertices of the polygon (ii) Mobile guards: Guards can move on the edges of the polygon (iii) Restriction on the visibility: The visibility of the guards can be restricted to 180o, 120o , 90o , etc. (iv) The art gallery can be curved too (Figure 5), which can be visualized as a curvilinear art gallery, that can be with inner holes too (Figure 6).

The important references on research on variants and relatives of art gallery problem are (i) O'Rourke J.(1987), Art Gallery Theorems and Algorithms, Oxford University Press, Inc. New York, NY, USA and (ii) N. Xu, Complexity of minimum corridor guarding problems, Information Processing Letters, vol. 112, no. 17-18, pp. 691696, 2012.

Figure 5: Hyndman gallery in Lubeznik center for the Arts, where the walls are curved

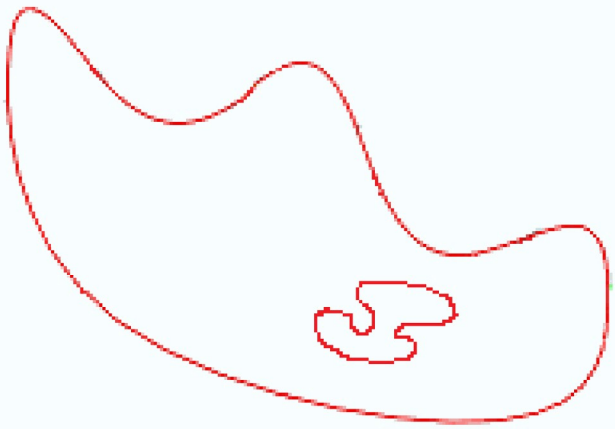


Figure 6: A curvilinear art gallery with holes

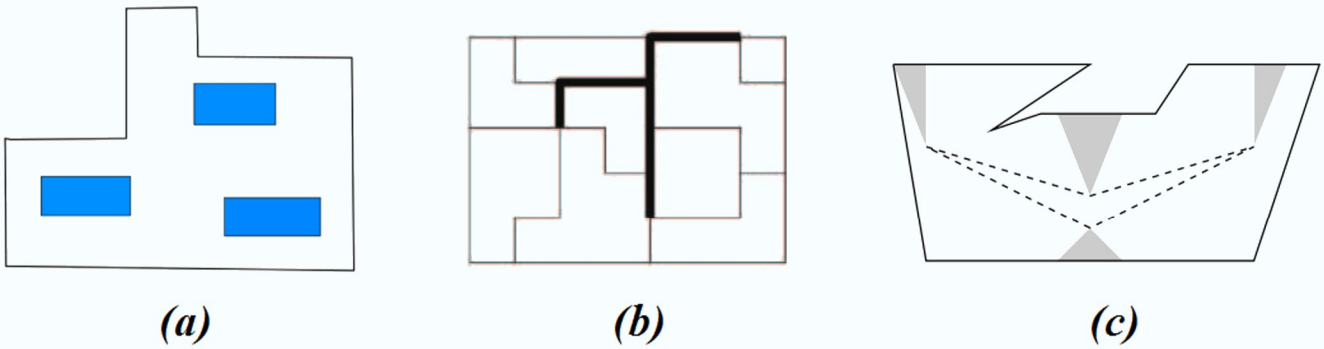
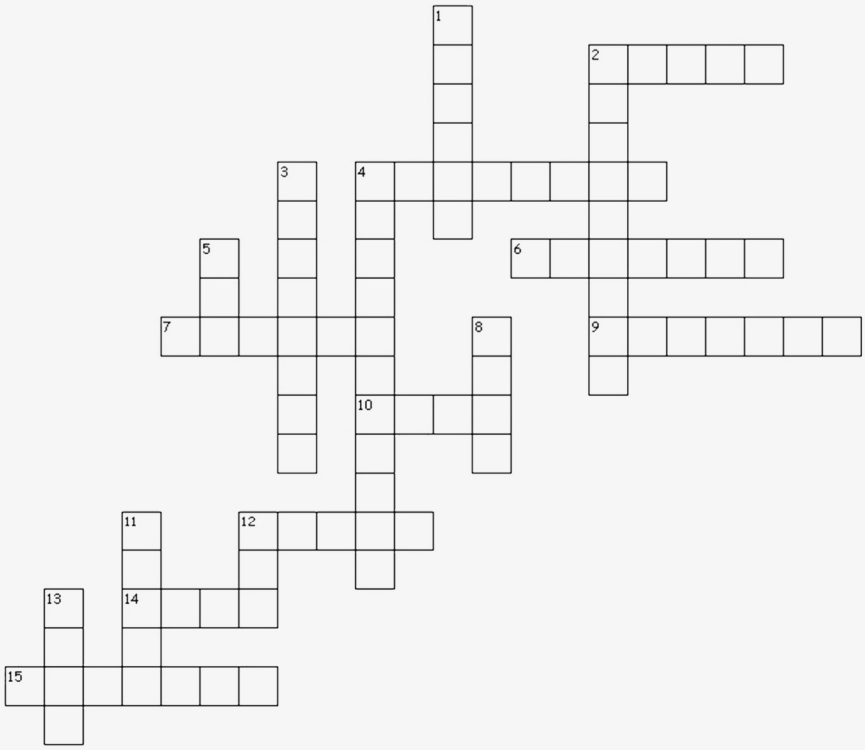


Figure 7: (a) Polygon with holes (shaded in blue) (b) Rectilinear polygon with minimum length tree (in bold lines) incident to all the partitions (c) Polygon with convex disjoint cages (shaded in grey) and a tour visiting all the cages (in dotted lines)

SEPTEMBER CROSSWORD



RULES

1. Email your solutions to csea@nitc.ac.in with "Threads September Crossword" as the subject. Screenshots are accepted.

2. The winner will receive a giftcard worth Rs. 200.

3. Only emails from a valid nitc email will be considered.

4. The first one to send in the correct solution to the crossword will be declared as the winner.

5. The winner will be contacted and the giftcard will be handed over within fourteen working days.

6. The decision of the team will be final.

ACROSS

2. Human:Eat::Computer:?

4. Touching the intangible is hard, where ever you are.

6. When you don't need some lines of code but don't wish to remove them...

7. for i.....: for j.....: do something;

8. All nodes say 'Hi' to each other.

10. One of the several species of fishes of the genus Salvelinus. Also a data type.

12. How to get out of while(1)

14. ... << 'This isn't left shift'

15. ... and then define.

DOWN

1. There are 10 types of people in the world...

2. In short, plus plus.

3. Procedure is key to do work. 12 tribes on your keyboard.

4. 3499 = DAB (on them)

5. "Terminal is too complicated, just give me everything in one GUI."

8. 4 bytes is one...

11. "Can't get out of this block because I'm a ..."

12. Atoms to the data.(Suarez, not again.)

13. Don't forget to return 0 in the end.

CODINGCALENDAR

September

Sept 21	Codeforces Round 511 (Div 1 & 2)
Hackathon with google India	Sept 22
Sept 23	Codechef September Cookoff
Zhejiang Lab Cup Global AI Competition (registration deadline)	Sept 25
Sept 29	Codechef September Lunchtime

October

HourRank 30	Oct 2
Oct 5	Goldman Sachs Women's CodeSprint
Codechef October Long Challenge	Oct 5
Oct 12	Moody's Analytics Women in Engineering Hackathon 2018
Codechef Snackdown 19 (registration ends)	Oct 16