Algo Made Easy

Pratima Pal[1], Kushagra Bhadauria[2], Nikita Patel[3], Kashif Akhtar[4], Apoorv Mishra[5]

Department of Computer Science And Engineering, [1,3,4]Maharana Institute of Professional Studies,
[2] Maharana Pratap Engineering College, Kanpur, Uttar Pradesh, India 209217

Abstract-

Competitive Programming is a sport, I mean literally. Take any sport, let's consider cricket for that matter, you walk in to bat for the first time. Swing and a miss, do it couple of times and you'll eventually hit one over the ropes. Now, consider a programming contest as a game of cricket, metaphorically. Compile a code and submit, you may get a WA (Wrong Answer). Make changes to code and eventually you will get your first AC (Accepted/Correct Answer).Let me give you a sneak peek, about 20% of questions in a programming contest are simple conversion of plain English to a code of your favourite programming language. Walk right into it, you will learn the unwritten rules of the game as you play harder and get better. And believe me, you don't need to know any "fancy name" to get started. Simply the basic understanding of Data Structures and algorithms will work fine. Ever heard of "Waft shot", yet you're the best batsman in your street, right? To move forward as competitive programmer, Data Structures and Algorithms will be your long-term partner who will help you solve problems with low time and space complexities. Thus the main objective of this project is to provide students the basic understanding of how Data Structure and Algorithm works.

INTRODUCTION

A GUI based desktop app made using Tkinter and Pygame modules of Python to visualize different computer algorithms(Searching, Sorting and Backtracking).
It introduces various common algorithms that are useful for a CSE department student to begin the journey in the most interesting subject on earth i.e. Data Structures.It is a boredom task to learn these algorithms by on a blackboard.How about learning it in a graphical way through animations?

This will make it easier for one to understand the logic in a very efficient manner.Also they will be able to determine the time complexity and auxiliary space required for each algorithm.

One of the most common problems in computer science education is the way to explain and communicate non-material terms and conditions. As a result the complexity increases when the concept is not visible, as it is fundamental in computer science, such as

algorithms. Various methods of support for teaching algorithms have been proposed, including the use of graphical and oral presentations of algorithms. The beautiful AVs make the algorithms come to life by clearly representing their various regions and highlighting the transformation between those regions. They display data structures in natural, invisible ways instead of focusing on memory addresses and activity calls. AVs are naturally appealing to teachers, almost worldwide who look good and are consistently “liked” by students. They can be used to attract students' attention during lectures, explain ideas in concrete terms, promote a practical learning process, and facilitate better communication between students and teachers. Interactive algorithm viewing allows students to evaluate and evaluate ideas regarding their individual needs. The purpose of this paper is to review the Visualization Algorithm textbooks, highlighting the work done so far in this area and the scope of development. We present the findings of the field of Visualization Algorithm (AV) based on our analysis. We regard the recognition of the algorithm as a priority and a point of view for further research and

By now you’ve realised that the questions are framed to deceive the way we think. Sometimes, If you convert plain english to code, you’d end up with TLE (Time Limit Exceeded) verdict. You need to learn a set of new techniques and algorithms to cope up with the time limits. In certain cases, Dynamic Programming (DP) comes to the rescue. Infact, you might have already intuitively used this technique. There’s always at least one question in any contest that can be solved by DP.

Thus to become a pro at this game. If you fail in one, go back and analyze where you went wrong and this Algo Made Easy will help you in mastering those concepts where you lack.

II. BLOCKCHAIN ARCHITECTURE

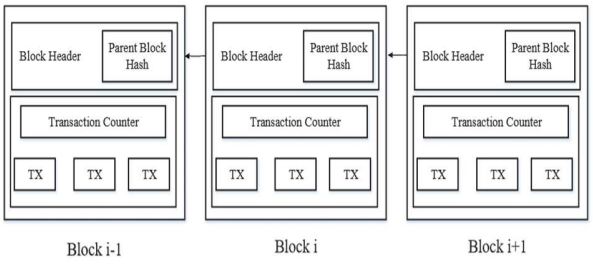

Fig. 1: An example of blockchain which consists of a continuous sequence of blocks.

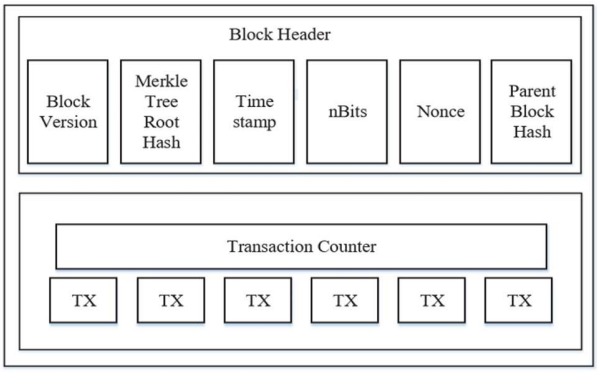

Fig. 2: Block structure

An algorithm is considered efficient if its resource consumption, also known as computational cost, isat or below some acceptable level. Roughly speaking, 'acceptable' means: it will run in a reasonable amount of time or space on an available computer, typically as a function of the size of the input.

Since the 1950s computers have seen dramatic increases in both the available computational powerand in the available amount of memory, so current acceptable levels would have been unacceptableeven 10 years ago. In fact, thanks to the approximate doubling of computer power every 2 years, tasks that are acceptably efficient on modern smartphones and embedded systems may have been unacceptably inefficient for industrial servers 10 years ago

Blockchain is a sequence of blocks, which holds a complete list of transaction records

like conventional public ledger [14]. Figure 1 illustrates an example of a blockchain. With a previous block hash contained in the block header, a block has only one *parent block*. It is worth noting that *uncle blocks* (children of the block's ancestors) hashes would also be stored in ethereum blockchain [15]. The first block of a blockchain is called *genesis block* which has no parent block. We then explain the internals of blockchain in details.

A. Block

A block consists of the *block header* and the *block body* as shown in Figure 2. In particular, the block header includes:

(i) Block version: indicates which set of block validationrules to follow.

(ii) Merkle tree root hash: the hash value of all the transactions in the block.

(iii) Timestamp: current time as seconds in universal timesince January 1, 1970.

(iv) nBits: target threshold of a valid block hash.

(v) Nonce: an 4-byte field, which usually starts with 0 and increases for every hash calculation (will be explained in details in Section III).

(vi) Parent block hash: a 256-bit hash value that points tothe previous block.

The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions [13]. Digital signature based on asymmetric cryptography is used in an untrustworthy environment. We next briefly illustrate digital signature.

B. Digital Signature

Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digital signed transactions are broadcasted throughout the whole network. The typical digital signature is involved with two phases: *signing phase* and *verification phase*. For instance, an user Alice wants to send another user Bob a message. (1) In the signing phase, Alice encrypts her data with her private key and sends Bob the encrypted result and original data. (2) In the verification phase, Bob validates the value with Alice's public key. In that way, Bob could easily check if the data has been tampered or not. The typical digital signature algorithm used in blockchains is the elliptic curve digital signature algorithm (ECDSA) [16].

C. Key Characteristics of Blockchain

In summary, blockchain has following key characteristics.

1. ***Decentralization***. In conventional centralized transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank), inevitably resulting to the cost and the performance bottlenecks at the central servers. Contrast to the centralized mode, third party is no longer needed in blockchain. Consensus algorithms in blockchain are used to maintain data consistency in distributed network.

2. ***Persistency***. Transactions can be validated quickly and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or rollback transactions once they are included in the blockchain. Blocks that contain invalid transactions could be discovered immediately.

3. ***Anonymity***. Each user can interact with the blockchain with a generated address, which does not reveal the real identity of the user. Note that blockchain cannot guarantee the perfect privacy

preservation due to the intrinsic constraint (details will be discussed in section IV).

comparison among the three types of blockchains is listed in Table I.

TABLE I: Comparisons among *public blockchain*, *consortium blockchain* and *private blockchain*

Property	Public blockchain	Consortium blockchain	Private blockchain
Consensus determination	All miners	Selected set of nodes	One organization
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus process	Permissionless	Permissioned	Permissioned

4. Auditability. Bitcoin blockchain stores data about user balances based on the Unspent Transaction Output (UTXO) model [2]: Any transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the state of those referred unspent transactions switch from unspent to spent. So transactions could be easily verified and tracked.

D. Taxonomy of blockchain systems

Current blockchain systems are categorized roughly into three types: public blockchain, private blockchain and consortium blockchain [17]. In public blockchain, all records are visible to the public and everyone could take part in the consensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium blockchain. As for private blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process.

A private blockchain is regarded as a centralized network since it is fully controlled by one organization. The consortium blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus. The

1.Consensus determination. In public blockchain, each node could take part in the consensus process. And only a selected set of nodes are responsible for validating the block in consortium blockchain. As for private chain, it is fully controlled by one organization and the organization could determine the final consensus.

2.Read permission. Transactions in a public blockchain are visible to the public while it depends when it comes to a private blockchain or a consortium blockchain.

3.Immutability. Since records are stored on a large number of participants, it is nearly impossible to tamper transactions in a public blockchain. Differently, transactions in a private blockchain or a consortium blockchain could be tampered easily as there are only limited number of participants.

4.Efficiency. It takes plenty of time to propagate transactions and blocks as there are a large number of nodes on public blockchain network. As a result, transaction throughput is limited and the latency is high. With fewer validators, consortium blockchain and private blockchain could be more efficient.

5.Centralized. The main difference among the three types of blockchains is that public blockchain is decentralized, consortium blockchain is partially centralized and private blockchain is fully centralized as it is controlled by a single group.

6.Consensus process. Everyone in the world could join the consensus process of the public blockchain. Different from public blockchain, both consortium blockchain and private blockchain are permissioned.

Since public blockchain is open to the world, it can attract many users and communities are active. Many public blockchains emerge day by day. As for consortium blockchain, it could be applied into many business applications. Currently Hyperledger [18] is developing business consortium blockchain frameworks. Ethereum also has provided tools for building consortium blockchains [19].

III. CONSENSUS ALGORITHMS

In blockchain, how to reach consensus among the untrustworthy nodes is a transformation of the Byzantine Generals (BG) Problem, which was raised in [20]. In BG problem, a group of generals who command a portion of Byzantine army circle the city. Some generals prefer to attack while other generals prefer to retreat. However, the attack would fail if only part of the generals attack the city. Thus, they have to reach an agreement to attack or retreat. How to reach a consensus in distributed environment is a challenge. It is also a challenge for blockchain as the blockchain network is distributed. In blockchain, there is no central node that ensures ledgers on distributed nodes are all the same. Some protocols are needed to ensure ledgers in different nodes are consistent. We next present several common approaches to reach a consensus in blockchain.

A. Approaches to consensus

PoW (Proof of work) is a consensus strategy used in the Bitcoin network [2]. In a decentralized network, someone has to be selected to record the transactions. The easiest way is random selection. However, random selection is vulnerable to attacks. So if a node wants to publish a block of transactions, a lot of work has to be done to prove that the node is not likely to attack the network. Generally the work means computer

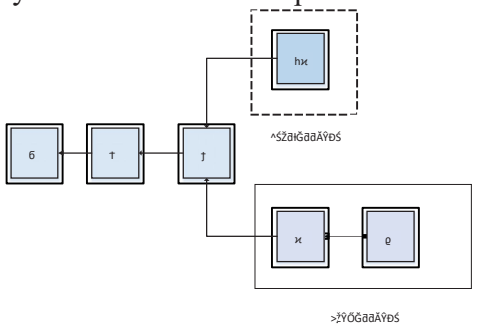

Fig. 3: An scenario of blockchain branches (the longer branch would be admitted as the main chain while the shorter one would be deserted)

calculations. In PoW, each node of the network is calculating a hash value of the block header. The block header contains a nonce and miners would change the nonce frequently to get different hash values. The consensus requires that the calculated value must be equal to or smaller than a certain given value. When one node reaches the target value, it would broadcast the block to other nodes and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other miners would append this new block to their own blockchains. Nodes that calculate the hash values are called *miners* and the PoW procedure is called *mining* in Bitcoin.

In the decentralized network, valid blocks might be generated simultaneously when multiple nodes find the suitable nonce nearly at the same time. As a result, branches may be generated as shown in Figure 3. However, it is

Property	PoW	PoS	PBFT	DPOS	Ripple	Tendermint
Node identity management	open	open	permissioned	open	Open	permissioned
Energy saving	no	partial	yes	partial	Yes	yes
Tolerated power of adversary	<25% computing power	<51% stake	<33.3% faulty replicas	<51% validators	<20% faulty nodes in UNL	<33.3% byzantine voting power
Example	Bitcoin [2]	Peercoin [21]	Hyperledger Fabric [18]	Bitshares [22]	Ripple [23]	Tendermint [24]

unlikely that two competing forks will generate next block simultaneously. In PoW protocol, a chain that becomes longer thereafter is judged as the authentic one. Consider two forks created by simultaneously validated blocks U4 and B4. Miners keep mining their blocks until a longer branch is found. B4,B5 forms a longer chain, so the miners on U4 would switch to the longer branch.

Miners have to do a lot of computer calculations in PoW, yet these works waste too much resources. To mitigate the loss, some PoW protocols in which works could have some side-applications have been designed. For example, Primecoin [25] searches for special prime number chains which can be used for mathematical research.

PoS (Proof of stake) is an energy-saving alternative to PoW. Miners in PoS have to prove the ownership of the amount of currency. It is believed that people with more currencies would be less likely to attack the network. The selection based on account balance is quite unfair because the single richest person is bound to be dominant in the network. As a result, many solutions are proposed with the combination of the stake size to decide which one to forge the next block. In particular, Blackcoin [26] uses randomization to predict the next generator. It uses a formula that looks for the lowest hash value in combination with the size of the stake. Peercoin [21] favors coin age based selection. In Peercoin, older and larger sets of coins have a greater probability of mining the next block. Compared to PoW, PoS saves more energy and is more effective.

Unfortunately, as the mining cost is nearly zero, attacks might come as a consequence. Many blockchains adopt PoW at the beginning and transform to PoS gradually. For instance, ethereum is planing to move from Ethash (a kind of PoW) [27] to Casper (a kind of PoS) [28].

PBFT(Practical byzantine fault tolerance) is a replication algorithm to tolerate byzantine faults [29]. Hyperledger Fabric [18] utilizes the PBFT as its consensus algorithm since PBFT could handle up to 1/3 malicious byzantine replicas. A new block is determined in a round. In each round, a primary would be selected according to some rules. And it is responsible for ordering the transaction. The whole process could be divided into three phase: *pre-prepared*, *prepared* and *commit*. In each phase, a node would enter next phase if it has received votes from over 2/3 of all nodes. So PBFT requires that every node is known to the network. Like PBFT, Stellar Consensus Protocol (SCP) [30] is also a Byzantine agreement protocol. In PBFT, each node has to query other nodes while SCP gives participants the right to choose which set of other participants to believe. Based on PBFT, Antshares [31] has implemented their dBFT (delegated byzantine fault tolerance). In dBFT, some professional nodes are voted to record the transactions.

DPOS(Delegated proof of stake). The major difference between PoS and DPOS is that PoS is direct democratic while DPOS is representative democratic. Stakeholders elect their delegates to generate and validate blocks.

With significantly fewer nodes to validate the block, the block could be confirmed

quickly, leading to the quick confirmation of transactions. Meanwhile, the parameters of the network such as *block size* and *block intervals* could be tuned by delegates. Additionally, users need not to worry about the dishonest delegates as they could be voted out easily. DPOS is the backbone of Bitshares[22].

Ripple [23] is a consensus algorithm that utilizes collectively-trusted subnetworks within the larger network. In the network, nodes are divided into two types: *server* for participating consensus process and *client* for only transferring funds. Each server has an Unique Node List (UNL). UNL is important to the server. When determining whether to put a transaction into the ledger, the server would query the nodes in UNL and if the received agreements have reached 80%, the transaction would be packed into the ledger. For a node, the ledger will remain correct as long as the percentage of faulty nodes in UNL is less than 20%.

Tendermint [24] is a byzantine consensus algorithm. A new block is determined in a round. A proposer would be selected to broadcast an unconfirmed block in this round. It could be divided into three steps: 1) *Prevote step*. Validators choose whether to broadcast a prevote for the proposed block. 2) *Precommit step*. If the node has received more than 2/3 of prevotes on the proposed block, it broadcasts a precommit for that block. If the node has received over 2/3 of precommits, it enters the commit step. 3) *Commit step*. The node validates the block and broadcasts a commit for that block. if the node has received 2/3 of the commits, it accepts the block. Contrast to PBFT, nodes have to lock their coins to become validators. Once a validator is found to be dishonest, it would be punished.

B. Consensus algorithms comparison

Different consensus algorithms have different advantages and disadvantages. Table II gives a comparison between different consensus algorithms and we use the properties given by [32].

1.Node identity management. PBFT needs to know the identity of each miner in order to select a primary in every round while Tendermint needs to know the validators in order to select a proposer in each round. For PoW, PoS, DPOS and Ripple, nodes could join the network freely.

2.Energy saving. In PoW, miners hash the block header continuously to reach the target value. As a result, the amount of electricity required to process has reach an immense scale. As for PoS and DPOS, miners still have to hash the block header to search the target value but the work has been largely reduced as the search space is designed to be limited. As for PBFT, Ripple and Tendermint, there is no mining in consensus process. So it saves energy greatly.

3.Tolerated power of adversary. Generally 51% of hash power is regarded as the threshold for one to gain control of the network. But selfish mining strategy [10] in PoW systems could help miners to gain more revenue by only 25% of the hashing power. PBFT and Tendermint is designed to handle up to 1/3 faulty nodes. Ripple is proved to maintain correctness if the faulty nodes in an UNL is less than 20%.

4.Example. Bitcoin is based on PoW while Peercoin is a new peer-to-peer PoS cryptocurrency. Further, Hyperledger Fabric utilizes PBFT to reach consensus. Bitshares, a smart contract platform, adopts DPOS as their consensus algorithm. Ripple implements the Ripple protocol while Tendermint devises the Tendermint protocol.

PBFT and Tendermint are permissioned protocols. Node identities are expected to be known to the whole network, so they might be used in commercial mode rather than public. PoW and PoS are suitable for public blockchain. Consortium or private blockchain might has preference for PBFT, Tendermint, DPOS and Ripple.

C. Advances on consensus algorithms

A good consensus algorithm means efficiency, safty and convenience. Recently, a number of endeavors have been made to improve consensus algorithms in blockchain. New consensus algorithms are devised aiming to solve some specific problems of blockchain. The main idea of PeerCensus [33] is to decouple block creation and transaction confirmation so that the consensus speed can be significantly increased. Besides, Kraft [34] proposed a new consensus method to ensure that a block is generated in a relatively stable speed. It is known that high blocks generation rate compromise Bitcoin's security. So the Greedy Heaviest-Observed Sub-Tree (GHOST) chain selection rule [35] is proposed to solve this problem. Instead of the longest branch scheme, GHOST weights the branches and miners could choose the better one to follow. Chepurnoy et al. [36] presented a new consensus algorithm for peer-topeer blockchain systems where anyone who provides noninteractive proofs of retrievability for the past state snapshots is agreed to generate the block. In such a protocol, miners only have to store old block headers instead of full blocks.

IV. CHALLENGES & RECENT ADVANCES

Despite the great potential of blockchain, it faces numerous challenges, which limit the wide usage of blockchain. We enumerate some major challenges and recent advances as follows.

A. Scalability

With the amount of transactions increasing day by day, the blockchain becomes bulky. Each node has to store all transactions to validate them on the blockchain because they have to check if the source of the current transaction is unspent or not. Besides, due to the original restriction of block size and the time interval used to generate a new block, the Bitcoin blockchain can only process nearly 7 transactions per second, which cannot fulfill the requirement of processing millions of transactions in real-time fashion. Meanwhile, as the capacity of blocks is very small, many small transactions might be delayed since miners prefer those transactions with high transaction fee.

There are a number of efforts proposed to address the scalability problem of blockchain, which could be categorized into two types:

• ***Storage optimization of blockchain***. Since it is harder for node to operate full copy of ledger, Bruce proposed a novel cryptocurrency scheme, in which the old transaction records are removed (or forgotten) by the network [37]. A database named account tree is used to hold the balance of all non-empty addresses. Besides lightweight client could also help fix this problem. A novel schem named VerSum [38] was proposed to provide another way allowing lightweight clients to exist. VerSum allows lightweight clients to outsource expensive computations over large inputs. It ensures the computation result is correct through comparing results from multiple servers.

• ***Redesigning blockchain***. In [39], Bitcoin-NG (Next Generation) was proposed. The main idea of Bitcoin-NG is to decouple conventional block into two parts: key block for leader election and microblock to store transactions. The protocol divides time into epoches. In each epoch, miners have to hash to generate a key block. Once the key block is generated, the node becomes the leader who is responsible for generating microblocks. Bitcoin-NG also extended the heaviest (longest) chain strategy in which microblocks carry no weight. In this way, blockchain is redesigned and the tradeoff between block size and network security has been addressed.

B. Privacy Leakage

Blockchain can preserve a certain amount of privacy through the public key and private key. Users transact with their private key and public key without any real identity exposure. However, it is shown in [40], [5] that blockchain

cannot guarantee the *transactional privacy* since the values of all transactions and balances for each public key are publicly visible. Besides, the recent study [41] has shown that a user's Bitcoin transactions can be linked to reveal user's information. Moreover, Biryukov et al. [11] presented an method to link user pseudonyms to IP addresses even when users are behind Network Address Translation (NAT) or firewalls. In [11], each client can be uniquely identified by a set of nodes it connects to. However, this set can be learned and used to find the origin of a transaction. Multiple methods have been proposed to improve anonymity of blockchain, which could be roughly categorized into two types:

1.Mixing [42]. In blockchain, users addresses are pseudonymous. But it is still possible to link addresses to user real identity as many users make transactions with the same address frequently. Mixing service is a kind of service which provides anonymity by transferring funds from multiple input addresses to multiple output addresses. For example, user Alice with address A wants to send some funds to Bob with address B. If Alice directly makes a transaction with input address A and output address B, relationship between Alice and Bob might be revealed. So Alice could send funds to a trusted intermediary Carol. Then Carol transfer funds to Bob with multiple inputs c1, c2, c3, etc., and multiple output d1, d2, B, d3, etc. Bob's address B is also contained in the output addresses. So it becomes harder to reveal relationship between Alice and Bob. However, the intermediary could be dishonest and reveal Alice and Bob's private information on purpose. It is also possible that Carol transfers Alice's funds to her own address instead of Bob's address. Mixcoin [43] provides a simple method to avoid dishonest behaviours. The intermediary encrypts users' requirements including funds amount and transfer date with its private key. Then if the intermediary did not transfer the money, anybody could verify that the intermediary cheated. However, theft is detected but still not prevented. Coinjoin [44] depends on a central mixing server to shuffle output addresses to prevent theft. And inspired by Coinjoin, CoinShuffle [45] uses decryption mixnets for address shuffling.

2.Anonymous. In Zerocoin [46], zero-knowledge proof is used. Miners do not have to validate a transaction with digital signature but to validate coins belong to a list of valid coins. Payment's origin are unlinked from transactions to prevent transaction graph analyses. But it still reveals payments' destination and amounts. Zerocash [47] was proposed to address this problem. In Zerocash, zero-knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARKs) is leveraged. Transaction amounts and the values of coins held by users are hidden.

C. Selfish Mining

Blockchain is susceptible to attacks of colluding selfish miners. In particular, Eyal and Sirer [10] showed that the network is vulnerable even if only a small portion of the hashing power is used to cheat. In selfish mining strategy, selfish miners keep their mined blocks without broadcasting and the private branch would be revealed to the public only if some requirements are satisfied. As the private branch is longer than the current public chain, it would be admitted by all miners. Before the private blockchain publishment, honest miners are wasting their resources on an useless branch while selfish miners are mining their private chain without competitors. So selfish miners tend to get more revenue.

Based on selfish mining, many other attacks have been proposed to show that blockchain is not so secure. In stubborn mining [48], miners could amplify its gain by non-trivially composing mining attacks with network-level eclipse attacks. The trail-stubbornness is one of the stubborn strategy that miners still mine the blocks even if the private chain is left behind. Yet in some cases, it can result in 13% gains in comparison with a non-

trail-stubborn counterpart. [49] shows that there are selfish mining strategies that earn more money and are profitable for smaller miners compared to simple selfish mining. But the gains are relatively small. Furthermore, it shows that attackers with less than 25% of the computational resources can still gain from selfish mining. To help fix the selfish mining problem, Heilman [50] presented an novel approach for honest miners to choose which branch to follow. With random beacons and timestamps, honest miners would select more fresh blocks. However, [50] is vulnerable to forgeable timestamps. ZeroBlock [51] builds on the simple scheme: Each block must be generated and accepted by the network within a maximum time interval. Within ZeroBlock, selfish miners cannot achieve more than its expected reward.

V. POSSIBLE FUTURE DIRECTIONS

Blockchain has shown its potential in industry and academia. We discuss possible future directions with respect to four areas: *blockchain testing*, *stop the tendency to centralization*, *big data analytics* and *blockchain application*.

A. Blockchain testing

Recently different kinds of blockchains appear and over 700 cryptocurrencies are listed in [52] up to now. However, some developers might falsify their blockchain performance to attract investors driven by the huge profit. Besides that, when users want to combine blockchain into business, they have to know which blockchain fits their requirements. So blockchain testing mechanism needs to be in place to test different blockchains.

Blockchain testing could be separated into two phases: *standardization phase* and *testing phase*. In standardization phase, all criteria have to be made and agreed. When a blockchain is born, it could be tested with the agreed criteria to valid if the blockchain works fine as developers claim. As for testing phase, blockchain testing needs to be performed with different criteria. For example, an user who is in charge of online retail business cares about the throughput of the blockchain, so the examination needs to test the average time from a user send a transaction to the transaction is packed into the blockchain, capacity for a blockchain block and etc.

B. Stop the tendency to centralization

Blockchain is designed as a decentralized system. However, there is a trend that miners are centralized in the mining pool. Up to now, the top 5 mining pools together owns larger than 51% of the total hash power in the Bitcoin network [53]. Apart from that, selfish mining strategy [10] showed that pools with over 25% of total computing power could get more revenue than fair share. Rational miners would be attracted into the selfish pool and finally the pool could easily exceed 51% of the total power. As the blockchain is not intended to serve a few organizations, some methods should be proposed to solve this problem.

C. Big data analytics

Blockchain could be well combined with big data. Here we roughly categorized the combination into two types: *data management* and *data analytics*. As for data management, blockchain could be used to store important data as it is distributed and secure. Blockchain could also ensure the data is original. For example, if blockchain is used to store patients health information, the information could not be tampered and it is hard to stole those private information. When it comes to data analytics, transactions on blockchain could be used for big data analytics. For example, user trading patterns might be extracted. Users can predict their potential partners' trading behaviours with the analysis.

D. Blockchain applications

Currently most blockchains are used in the financial domain, more and more applications for different fields are appearing. Traditional industries could take blockchain into consideration and apply blockchain into their fields to enhance their systems. For example, user reputations could be stored on blockchain. At the same time, the up-and-coming industry could make use of

blockchain to improve performance. For example, Arcade City [51], a ridesharing startup offers an open marketplace where riders connect directly with drivers by leveraging blockchain technology. A smart contract is a computerized transaction protocol that executes the terms of a contract [54]. It has been proposed for long time and now this concept can be implemented with blockchain. In blockchain, smart contract is a code fragment that could be executed by miners automatically. Smart contract has transformative potential in various fields like financial services and IoT.

VI. CONCLUSION

Many programmers argue that the problems in competitive programming do not relate to the real life programming work. For the most part, it is true. Then why do we do it? Because it makes you a better programmer. How? Time limit always makes you write time efficient solutions.

● Critical test data helps you write correct solutions, in one go!

● Further it makes you great at debugging code.

Hard problems makes you break down the problem into chunks, solve them individually and bring it all together to solve the main problem. Yes, competitive programming is not the only way to master these qualities but it is one of the best ways to do so. Give it a shot, if you enjoy it, it's worth it. If you don't, even after repeated trials, give yourself a break. And to get started with it DISA is the heart of programming and you can not ignore it while solving coding problems in competitive programming, Array Linked List, Stack, Queue, Tree, Trie, Graph, Sorting, Recursion, Dynamic Programming all these basic building blocks of DSA will help you to become a good programmer. The most important thing you need to know what, when and where to apply them. It means which data structure is suitable for what type of problem to get the optimal solution. LeamAlgo will help you in learning them in a best way possible.

VII. ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the Project undertaken during B.Tech, Final Year. We owe special debt of gratitude to our project supervisor Mr. Apoorv Mishra , Department of Computer Science and Engineering, Maharana Pratap Engineering College, Kanpur for his constant support and guidance throughout the course of our work. Her sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

VIII. REFRENCES

- Real Python https://realpython.com/python-gui-tkinter/
- Tutorials point https://www.tutorialspoint.com/python/python_gui_programming.htm
- Geeksforgeeks https://www.geeksforgeeks.org/python-gui-tkinter/
- W3schools https://www.w3schools.in/data-structures/tutorials/
- Javatpoint https://www.javatpoint.com/data-structure-tutorial