Collaborators: Brian C., Joselo D, RJ

Explanation of Cryptocurrency

The word “crypto” stems from cryptography which is “the art of writing and solving code”; fittingly, Cryptocurrency works the same way. Crypto allows for digital assets, may it be, nonfungible tokens, newly created currency, and countless more possibilities to be sent with extremely low transaction costs and rapidity. In our study, we refer to Cryptocurrency only in its use case of currency. Cryptocurrency is derived from blockchain technology, Blockchain technology is an operating system that stores transactions in the form of blocks of data that are eventually linked. Blockchain was created in the context of cryptocurrency to allow for safe data collection. Blockchain technology uses smart contracts to keep receipts of every single transaction but not the individual transferring the currency; meaning instead of having a log showcasing the individual's names and information when sending money (like traditional banks), they keep a log of the “wallet addresses" which are 42 character long codes to track wallets (which hold an individual's balance) sending each other Cryptocurrency. These wallets are then connected through a series of networks that are derived from the interactions each wallet has with others. The owner of these wallets can vary to being an individual or a corporate entity, but all these wallets when interacting with one another are connected through a P2P system. P2P stands for Peer-to-Peer, and what this system does is create a decentralized communication model where all parties and entities have the same capabilities to start an interaction. All the interactions that happen through the P2P system are encrypted, meaning that only the sender and the receiver of the transaction have access to specific information, such as how much money was traded. For all of these transactions to be legitimized and to be considered official they have to go through the process of Miners, which are nodes in the cryptocurrency networks which process any data that is being transacted and validate the block of data. These miners will create the blocks of data with the transaction to even further the blockchain. In some cases, the Miners can be composed of algorithms that process each transaction and save it to block data to validate all trades.

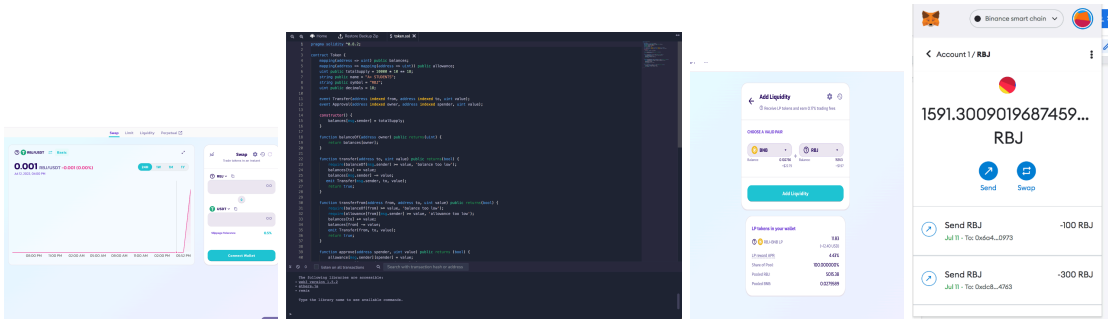
Why is cryptocurrency important

The cryptocurrency topic can be considered one of the most innovative movements in the free market economy. It has paved the way toward a networking system that is transparent and secure for the public. The incorporated P2P system is what allows cryptocurrencies to be able to be a public exchange system. Since cryptocurrency is not centralized or controlled by a government, it allows any individual to partake fairly in electronic commerce. To understand cryptocurrency is to understand how the world revolves around an economy, and by being able to be familiar with cryptocurrency any individual can take advantage of this and create independent economic growth. The importance of cryptocurrency is not only relevant economically but also socially, as many people have joined together to work for a single goal of creating and selling cryptocurrency. Cryptocurrency is made for the individual in the free market and it is perfectly designed to allow equality among investors which is why it is the commerce of the future, and it is always crucial to learn and adapt towards the future.

Our coin and its purpose

To further our study on Cryptocurrency and display the intricacy of a network via a crypto coin, we decided to create one; our coin is called “A+ students” with the ticker “RBJ”. Our coin is made to represent the possible simulation of a new coin, since we only have a few transactions in our coins network, the simulation serves to showcase 100 transactions to give us an idea of the possibility of our Cryptocurrency. The RBJ simulation was not made to show the network of blockchain but more the economical echo system of a coin with N amount of transaction (100 in our simulation) which is why we used a complex blockchain that works rather than a simple one we could explain. Each transaction is transferred from vertice to vertice via edge; each edge in our coin is weighted as the amount of the coins sent and each vertice as the wallet sending as well as how much the wallet receives and sends. RBJ was coded in solidity which is a coding language very similar to javascript on the “Binance smart chain”. The “Binance smart chain” is the second largest blockchain network that allows us to publish and code a coin in a fraction of a time as we are already using a perfectly well-running blockchain network. After coding RBJ we compiled it and simply published it on the smart chain so we could be publicly bought and sold and commence our coins network. To assign the coin any value we had to provide 10$ for the 10,000 coin supply meaning at the moment you were able to buy the whole supply for 10$. As we bought some of the supply and sent it to our wallets the price increased from 0.001$ per coin to 0.0014$ which proved to us we had a running network that adjusted to supply and demand. The discovery of this easily shifted price impact indicated that if a wallet were to sell all its supply the price would decrease and if they

would buy, the price would rise. What's preventing RBJ coins from losing all value in the simulation and even in practice is the fact that a wallet is only able to buy or sell under 15% of the token's full supply, so even though you could buy our whole supply for about 12$, you cannot buy or sell more than 1.8$ worth of our coin (15% of supply). In theory, however, if multiple people were to sell or buy the cascade effect would be in motion and our coin price would either decrease or increase but this is highly unlikely in practice or in a random simulation, which is how our coin was able to run a realistic simulation.



Comparison with larger cryptocurrency network

BNB, similar to RBJ, uses the Binance chain to exist and be transactable on the blockchain. In essence, as RBJ is built of BNB they work the same at a smaller scale. BNB is the native token of the Binance chain meaning it was created as the first token on the Binance chain. BNB is a proof of stake (PoS) token (like RBJ) meaning instead of using the traditional Proof of Work, which promotes a race to mine token by seeing which computer can solve the arithmetic code faster to be able to get a commission; PoS just uses algorithms to validate transactions that are powered by users of the network putting their coins up for collateral which provides liquidity in a liquidity pool. BNB found success after establishing itself as a superior network in transaction time and technical capabilities compared to other coins. BNB usually has a 4 million transaction in its 24-hour volume (how much is put in and out of the coin) - ycharts.com, which spans from hundreds to millions of wallets, meaning they're in and out-degree of the 163 million coin supply is moved heavily. Through our simulation, however, the max volume we could have moved is 10,000 as that is our coin supply meaning since we have such a low supply it would be highly unlikely we scale our coin to be able to make 4 million transactions a day and instead 100 is plenty (like shown in our network) and even then at the current price of our coin, wallets would only be

transacting a few cents a day. BNB can be staked for yield, this means not being able to sell your coins for a specific amount of time and you would be rewarded an APY (annual percentage yield) of 5-10%. Our coin on the contrary lacks this making our network inferior as you are not able to get money back after providing liquidity; this very concept is what's not allowing our coin to be scaled largely as there is barely any liquidity and buying and selling anything on the network would have too high of a price impact.

Sources:

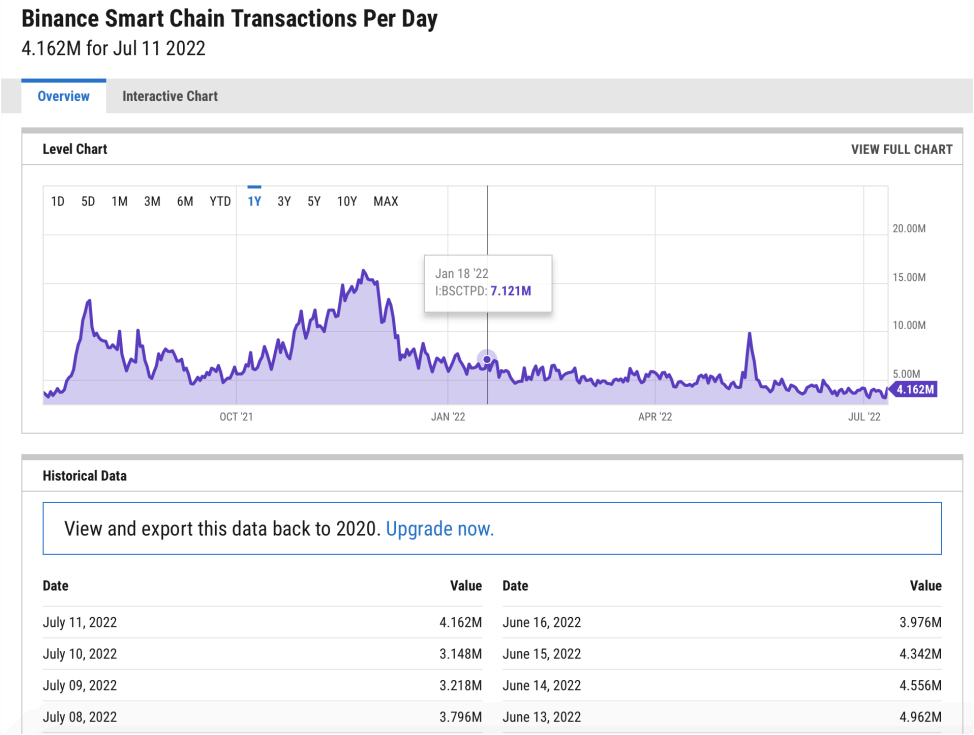


Fig. 1: The amount of transactions per day.

Importance of study

We initialized the study of cryptocurrency in networking to first analyze how money and transactions rotate through the interactions of individuals and entities. The simulation created was based on

Description of the model

How it works

The program will randomly generate a random directed weighted graph, and generate random transaction amounts as the weight of the edges (transactions). It will also create a ***# of nodes * 3*** matrix that simulates the blockchain of the network. The main goal of this program is to study the interactions within the P2P cryptocurrency network and discover the structure of the cryptocurrency network.

"Rand_Graph" Function

I made this function using the same idea as making an Erdo_Renyi graph, in which when the random probability is less than p (the input probability), the function makes an edge. The difference is that the output matrix is not symmetrical, because it's a directed graph. Instead of outputting a graph, this function outputs an adjacency matrix, and it'll be converted into a graph into the variable G.

"Unique_ID" Function

The purpose of this function is to generate unique IDs for each node, which is also known as the Wallet ID. In the real cryptocurrency network, it is being used to identify a wallet within the network. The input of this function will be the number of nodes in the network. "java. util.UUID.randomUUID" is a java method, which creates a UUID as a string, and it's stored in the result vector (ids). Then the result vector will be returned as the result of the function.

"Wallet_balance" Function

Each node in a cryptocurrency network contains a lot of information, one of them is the balance of the wallet. With the transaction being made within the network, the balance of the wallet is changing. However, if the balance of a wallet is less than 0, it will be removed from the network.

The input of this function would be the **graph** version of G, which is G1 (since G is being converted into a GraphPlot). We also input v, as the location of the wallet. "Balance" would be the balance of the wallet. The output would be the balance of the wallet i. We use functions "inedges" and

“outedges” to find the incoming and outgoing edges, and also their node index. For each outgoing edge, we decrease the balance by the weight of the edge, which is the amount of the transaction. We find the amount of each transaction by using the “findedge()” method, which finds the index of the edge, and we find the weight of the edge by accessing the weight property of graph G1.

Blockchain

The cryptocurrency blockchain is a shared, encrypted database that holds all the transactions that happened in the database. Each node in the network will have a copy of the blockchain. In this simulation, we create the blockchain by accessing the “Edges” property located in the digraph variable G1. There are two columns in the property, the first column is the end nodes of the edges, and the second column is the weights of the edges. After I find the index of the end nodes, I find the Wallet ID of them, which is located in the Wallet_ID vector. To have an n*3 matrix with two different types of elements, I created three column vectors: senders, receivers, and amounts. After plugging in values for each vector, I combined them to be an n*3 matrix called a blockchain, which is shown in the appendix.

The first column is the Wallet ID of the sender of the transaction, and the second column is the receiver's Wallet ID of the transaction. The third column is the number of coins traded in the transaction.

Resulting Network

In the resulting network, all of the nodes are the wallets that participated in this cryptocurrency network. The arrow on each edge would be the direction of the transaction. The numbers (weights) on the edges are the amount for each transaction. In the data tip box, in degree represents the number of incoming transactions of this wallet, and the out-degree would represent the number of outgoing transactions. ID is the Wallet ID of the wallet, which is helpful for the network to identify a node by its unique ID. Lastly, the balance is the remaining balance of the wallet after all the transactions.

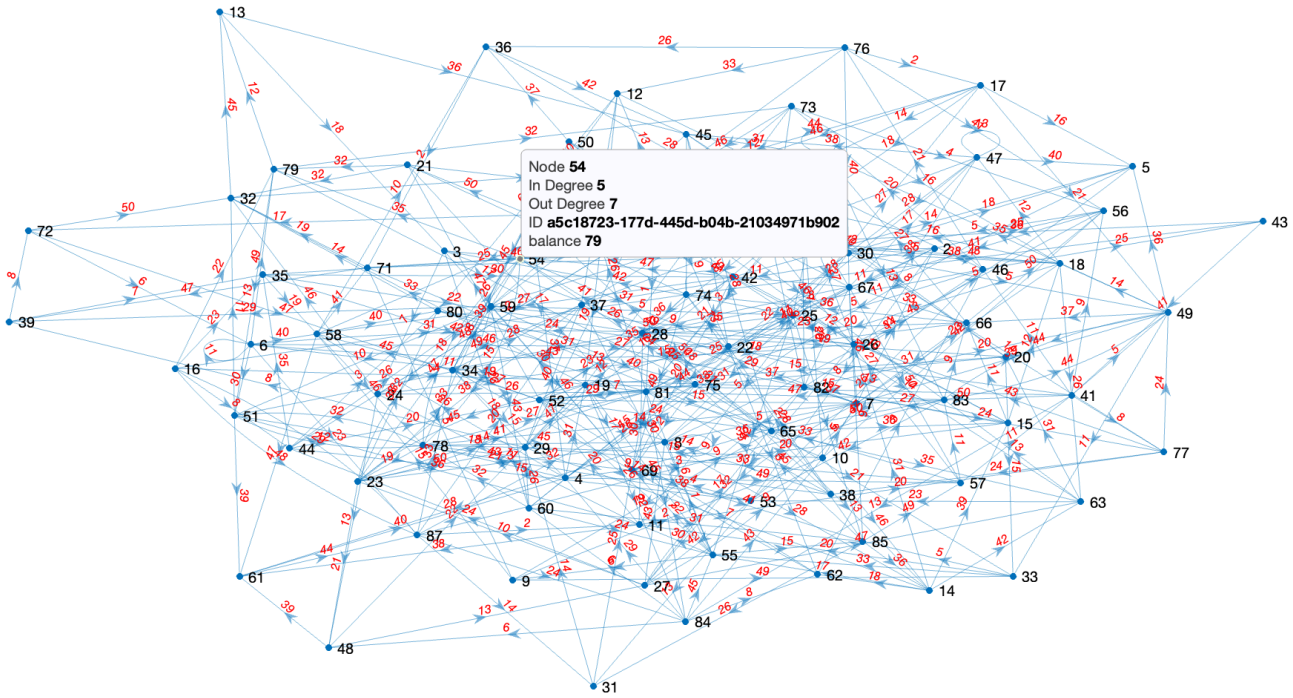


Fig. 2: The resulting network of the model.

Interpretation of simulation

Based on the results of the simulated graph there appeared to be a high dispersity of centrality, meaning that there was no vertex that could be considered as the central vertex as many vertices had the same amount of centrality. In the context of crypto currency networks this demonstrates that many wallets are going through many transactions hence why there is not a single vertex which is the “main " or central one. With this same denotation, that could be considered outliers with both very low centrality and high centrality could be defined as either wallets that do not participate much in the trading of crypto currency and wallets that are fully invested into having as many transactions as possible. These vertices with high centrality could be from wallets such as corporate entities or individuals who dedicate their work toward trading crypto. Since this is a simulation of a cryptocurrency network however, there shoudlt be a a central vertex as all wallets have the same opportunity to transact equally, even though there are wallets that participate more.

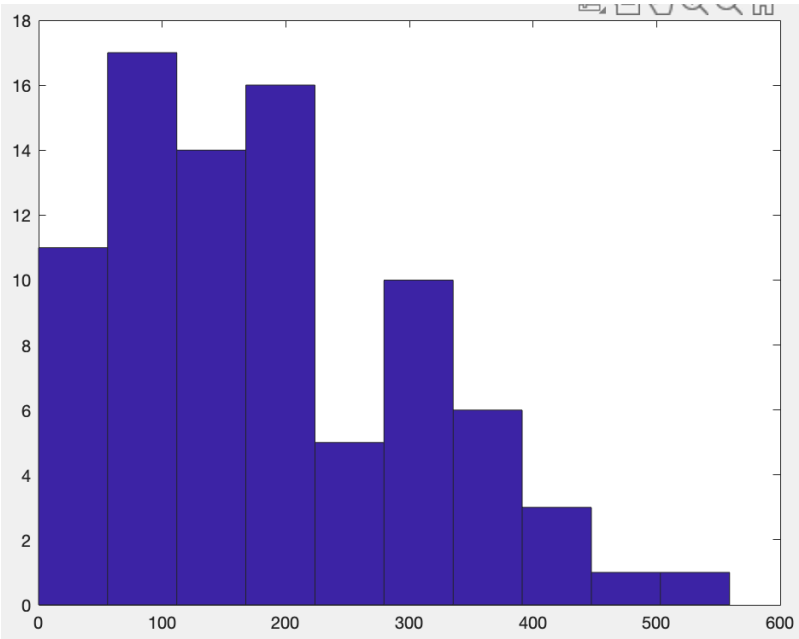
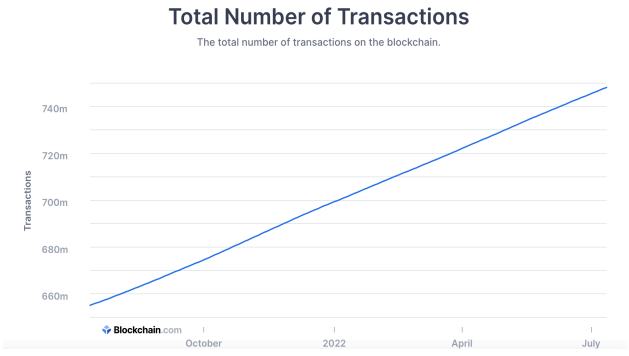


Fig. 3: X= betweenness centrality, Y= number of vertices

Comparison with real-world networks

In general, the statistics of a real-life cryptocurrency network consist of a generalized P2P system that tries to avoid centralization and allows each node to have as many connections as it desires. Bitcoin is the largest and first cryptocurrency that was able to implement a full P2P networking system which is why it is an exemplary network to follow as many of the current cryptocurrency networks follow its model. Bitcoin has the characteristics where decentralization does not exist as the network tries to include each node into the network equally. However, there are high levels of clustering with nodes and this is because, for the most part, the vertices do not have a parameter for how many connections there can be created, in Bitcoin the default parameter is set to be 125 connections per node but this could be changed by each user to many connections they wish to have. This, therefore, tested a high level of transitivity in cryptocurrency networks with nodes being able to create connections with any other node as long as they are in the same system. Specifically, in Bitcoin there are different networks with different parameters of the distances there can be between nodes. There is the “reachable Bitcoin network” which is composed of nodes that can have a range of connections between nodes of 5,000 to 10,000 (Segura et.al). There is also the “nonereachable Bitcoin” network which is composed of nodes that are in direct contact with the Bitcoin protocol and have a range of connections that is 10 times bigger than the reachable network. Lastly, there is the extended network which is composed of all the nodes in the Bitcoin ecosystem, meaning it includes vertices that are not directly involved with having transactions with other wallets but that deal with creating blocks of data, nodes like these are like the Miners. Since the Bitcoin system does not include any special nodes or unique

requirements for specific nodes, each vertex has an equal opportunity to create a connection hence why the overall statistic is a diagram of a cryptocurrency network will look with high transitive and displacement of nodes. In general, as well the increase in transactions and the creation of blocks of data within these networks is also defined through a linear relationship. Overall as time has passed there have been more transactions occurring in the cryptocurrency network, meaning transaction is increasing at a linear rate;



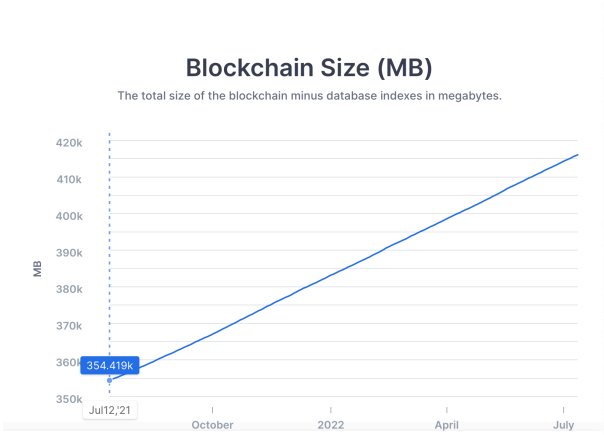


Fig. 4 Fig. 5

(Blockchain.com)

This statistic in cryptocurrency networks could be explained by how current nodes in a network are getting more interested in creating more transactions with different wallets, which is why it can be seen that nodes in a cryptocurrency network, over time, will create much more edges and connections. Based on the statistical characteristics of cryptocurrency networks such as Bitcoin, the model that we coded is very similar in the way that it allows for a clustering system but at the same time avoids any form of centralization. Since cryptocurrency networks find it very important to allow everyone to have an equal chance of creating connections, this is very important. The coded network as well deals with clustering of nodes and high transitive between these nodes which are highly exemplary of a cryptocurrency network. However, the simulated network is at a much smaller rate and parameters than a real-world cryptocurrency network as these work with high rates of transactions and many different nodes exchanging high amounts of money. The simulated diagram does not include nodes such as miners and mining pools as they could be considered as part of the extended network of a cryptocurrency ecosystem. Nevertheless, the simulated graph does give a good picture of how a cryptocurrency can disperse in a small network.

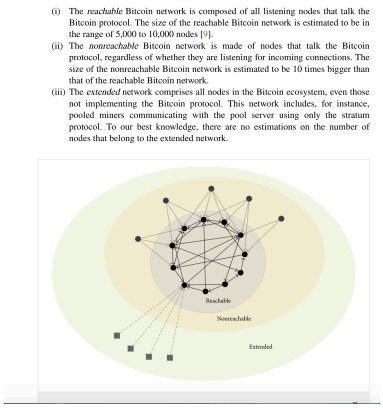


Fig. 6

Future inquiry:

Our coin, while functional, was imperfect and we could have added multiple features instead of running a simulation, actually legitimately making a network out of it. We could have used a program called legible to make thousands of wallets and randomly disperse our coins, so we could build our real network and then compare it to the results of MatLab. However, since our coin has such low value it would be complicated to transfer it to thousands of wallets, which is another pivotal point that could have been better; our coin lacks barely if any value so if we had more capital to provide for liquidity of the coin and our coin was worth more, instead of displaying the number of coins being transferred in the simulated network we can display the money moving around to more accurately depict an economical ecosystem. In coding the random graph we were not able to accurately display the blockchain as we weren't able to draw blocks, miners that validate transactions, and lacked a few of the principles of blockchain, instead of making an economical ecosystem with more time to work we could have made a functioning diagram of the Binance Smart blockchain in which or coin is based on.

Work Cited

Finance Research. “BNB (BNB).” *Finance Research*, Finance Research, 3 Sept. 2020, research.binance.com/en/projects/bnb.

Binance Smart Chain Transactions Per Day, ycharts.com/indicators/binance_smart_chain_transactions_per_day#:~:text=Basic%20Info,28.74%25%20from%20one%20year%20ago.

Delgado-Segura, Sergi, et al. “Cryptocurrency Networks: A New P2P Paradigm.” *Mobile Information Systems*, Hindawi, 1 Mar. 2018, www.hindawi.com/journals/misy/2018/2159082/.

Hayes, Adam. “Blockchain Explained.” *Investopedia*, Investopedia, 24 June 2022, www.investopedia.com/terms/b/blockchain.asp.

Kunapuli, Krishna. “How Cryptocurrency Is Redefining Networking.” *CXOToday.com*, 21 Sep. 2021, www.cxotoday.com/story/how-cryptocurrency-is-redefining-networking/#:~:text=In%20the%20cryptocurrency%20realm%2C%20peer-to-peer%20%28P2P%29%20networking%20is,operate%20on%20P2P%20networks%20comprising%20millions%20of%20nodes.

Kunapuli, Krishna. “How Cryptocurrency Is Redefining Networking.” *CXOToday.com*, 21 Sep. 2021, www.cxotoday.com/story/how-cryptocurrency-is-redefining-networking/#:~:text=In%20the%20cryptocurrency%20realm%2C%20peer-to-peer%20%28P2P%29%20networking%20is,operate%20on%20P2P%20networks%20comprising%20millions%20of%20nodes.

“Mastering Bitcoin.” *O'Reilly Online Learning*, O'Reilly Media, Inc., www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch06.html.

Satsuki, Pavel. “Types of Cryptocurrency Exchanges, Decentralized vs Centralized Exchanges.” *Soft*, Soft-FX, 10 June 2021, www.soft-fx.com/blog/types-of-cryptocurrency-exchanges/.

Sharma, Rakesh. “What Does the Bitcoin Blockchain Record?” *Investopedia*, Investopedia, 13 July 2022,

www.investopedia.com/ask/answers/063015/what-does-block-chain-record-bitcoin-exchange-transaction.asp#:~:text=Key%20 Takeaways,on%20the%20order%20of%20transactions.

“Blockchain.com.” *Blockchain.com*, www.blockchain.com/charts/blocks-size.

Appendix

Rand_Graph:

```
function RG = Rand_Graph(n, p)
% similar with ER Graph, but a directed graph
% OUTPUT: a adjacency matrix of a graph
    A = zeros(n);
    for i = 1:n
        for j = 1:n
            R = rand();
            if R < p
                A(i,j) = 1;
            end
        end
    end
    RG = A;
end
```

Unique_ID:

```
function id = Unique_ID(n)
    % OUTPUT: a vector with all the IDs for each node of a graph
    ids = strings(1,n);
    for i = 1:n
        temp =  java.util.UUID.randomUUID;
        myuuid = temp.toString;
        ids(i) = myuuid;
    end
    id = ids;
end
```

Wallet_balance:

```
function B = wallet_balance(G1, v, balance1)
    % G1 as the network as a GRAPH
    % v as the vertex
    % balance as the balance of the vertex v
    % OUTPUT: the balance for vertex i
    [out_e, receivers] = outedges(G1,v); % find the out edges
    [in_e, senders] = inedges(G1, v); % find the in edges
    % Reciever and senders are two vectors that stores the nodes of in &
    % out edges

    % minus all the outgoing transactions
    for i = 1:length(out_e)
        balance1 = balance1 - G1.Edges.Weight(findedge(G1,v,receivers(i)));
    end

    % adding all the incoming transactions
    for i = 1:length(in_e)
        balance1 = balance1 + G1.Edges.Weight(findedge(G1, senders(i), v));
    end
    B = balance1;
end
```

Blockchain code:

```
% Blockchain of the network
num_edges = numedges(G1);
senders = strings(num_edges, 1);
receivers = strings(num_edges, 1);
amount = G1.Edges.Weight;
for i = 1:num_edges
    s = G1.Edges.EndNodes(i, 1); % accesing the first column of EndNodes
    senders(i) = Wallet_ID(s);
    r = G1.Edges.EndNodes(i, 2); % accesing the second column of EndNodes
    receivers(i) = Wallet_ID(r);
end
blockchain = [senders, receivers, amount];  % a cell matrix with sender, r
```

Blockchain database:

2e83c776-0e40-...	0554a603-bed8-...	14
2e83c776-0e40-...	61bb1df7-8f21-...	25
c6049c96-5cc2-...	e0c172c1-c5f4-...	18
c6049c96-5cc2-...	05d093d1-cbc1-...	45
b455472c-499d...	d8e28a83-fe7a-...	20
b455472c-499d...	381f1b18-e7dd-...	33
b455472c-499d...	7d357820-53e6...	5
fb8f8fbc-6404-4...	8c40e784-3161-...	15
fb8f8fbc-6404-4...	e385636b-279e-...	28
fb8f8fbc-6404-4...	263c2c08-cc85-...	26
fb8f8fbc-6404-4...	5e4dfa80-5020-...	27
fb8f8fbc-6404-4...	3c255bdf-97e4-...	45
fb8f8fbc-6404-4...	7a6051e7-d8df-...	50
d4651ab9-792f-...	fee03e2e-a7b6-...	22