GHOST PROTOCOL

Blockchain technologies, lecture 3

Course overview

- Transaction pool and scaling
- Forks and transaction ordering
- Longest chain rule
- GHOST protocol

TRANSACTION POOL

Transaction pool

- New transactions are pending transactions.
- Before confirmation pending transactions are gathered in mempool.
- Nodes share mempool data by relaying transactions until it reaches the entire network.
- Transactions are transmitted either by the initiator of the transaction or when a node hears about a new transaction.
- Memory pool (Bitcoin)/TX-QUEUE (Parity)/TX-POOL (Geth) virtual waiting room collecting valid pending transactions until a miner processes them.

Transaction pool

- Each node maintains its own mempool, each node has its own storage capacity for unconfirmed transactions.
- When a transaction is confirmed, and included in the block, it is removed from mempool.
- Nodes prioritize transactions judging on transactions fees.
- Mem pool size can fluctuate as it depends on the number of transactions that are relayed.
- Heavy transaction volume increase delays in transaction confirmation time.
- 4.6 TPS -- visa 1700 TPS

Transaction pool, scaling

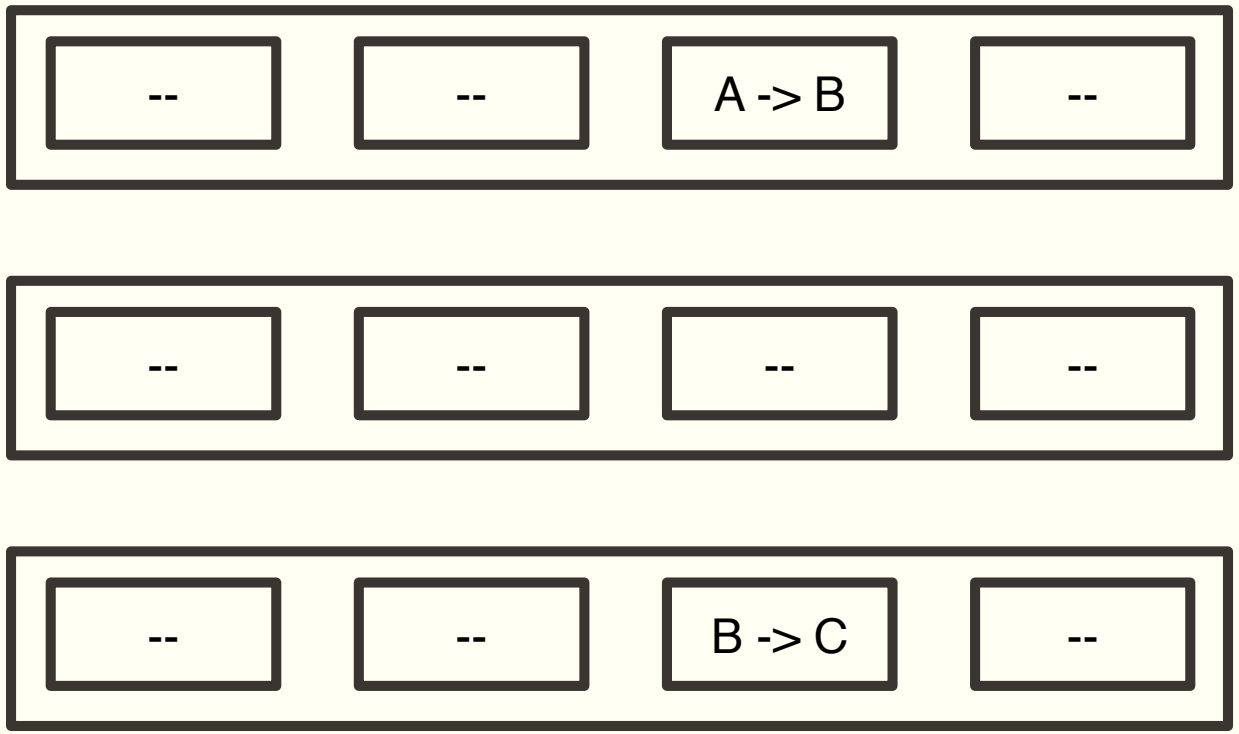
- Layer-2 scaling solutions are solutions designed to scale transaction processing capability by handling transactions off the mainnet playing the role of an arbitrator.
- State channels (Bitcoin Lighting network) based on payment channels
  - Protocol between a fix set of participants (often two).
  - Transactions between participants are securely processed off-chain.
  - Participants deposit funds with a 2-of-2 multisig transaction. For instance, Alice and Bob hold 5+5BTC on lightning channel.
  - After money is deposited both participants can send each other money without interaction with the mainchain.
  - Transactions are validated in the mainchain as a single transaction.

Transaction pool, scaling

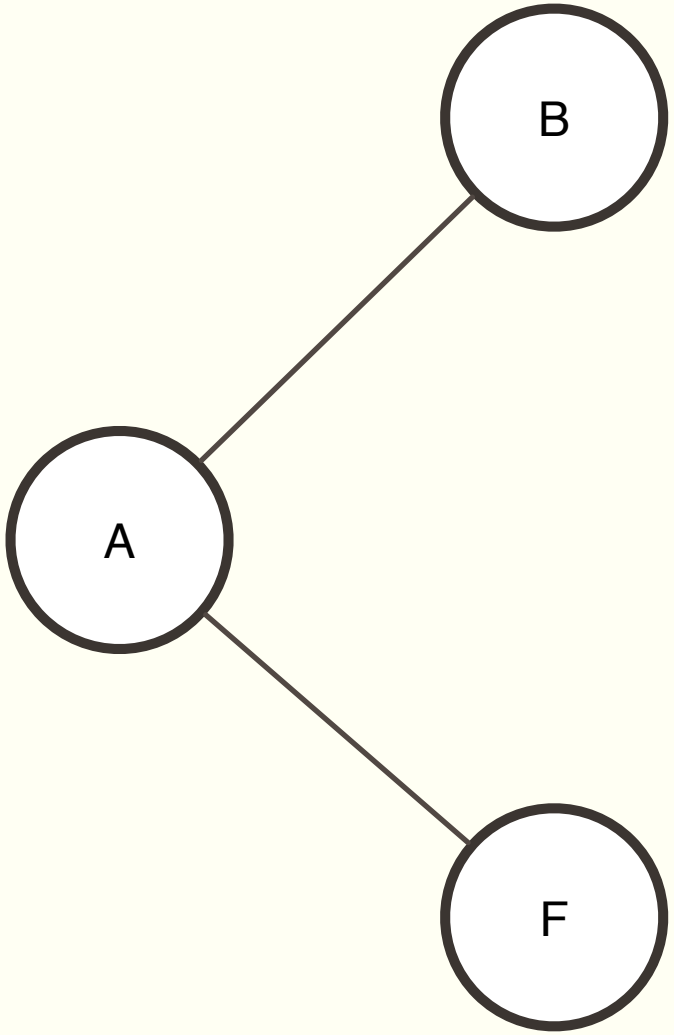
- Side chains (Ethereum, example Token Bridge)
  - Completely separate blockchain with its own consensus algorithm and its own validators.
  - Block headers are snapshotted to the mainchain.
  - Choice rule on side-channel: a block is canonical if it build on top of the latest snapshotted block.
  - Two-way-peg: transfer transactions cross-chain. Assets are locked on the mainchain and minted on side channel.
  - When assets are transferred back, they are burned from the side channel and unlocked on main channel.

Transaction pool, scaling

- Plasma Chash
  - Users deposit assets into the chain smart contract.
  - Plasma cash asset is represented by NFTs
  - Each Plasma block has a slot for every token. When a token is spent, a record of that transaction is placed at the corresponding slot.
  - Blocks in Plasma Cash form sparse Merkle Trees, providing proof that a token is not part of a specific block.
  - Users only keep information only about the tokens they own.
  - The proof of ownership consist of the full history of the token:
    - Owner after last transaction
    - Token wasn’t spent in another block.


--
--
A -> B
--
--
--
--
--
--
--
B -> C
--

FORKS AND TRANSACTION ORDERING

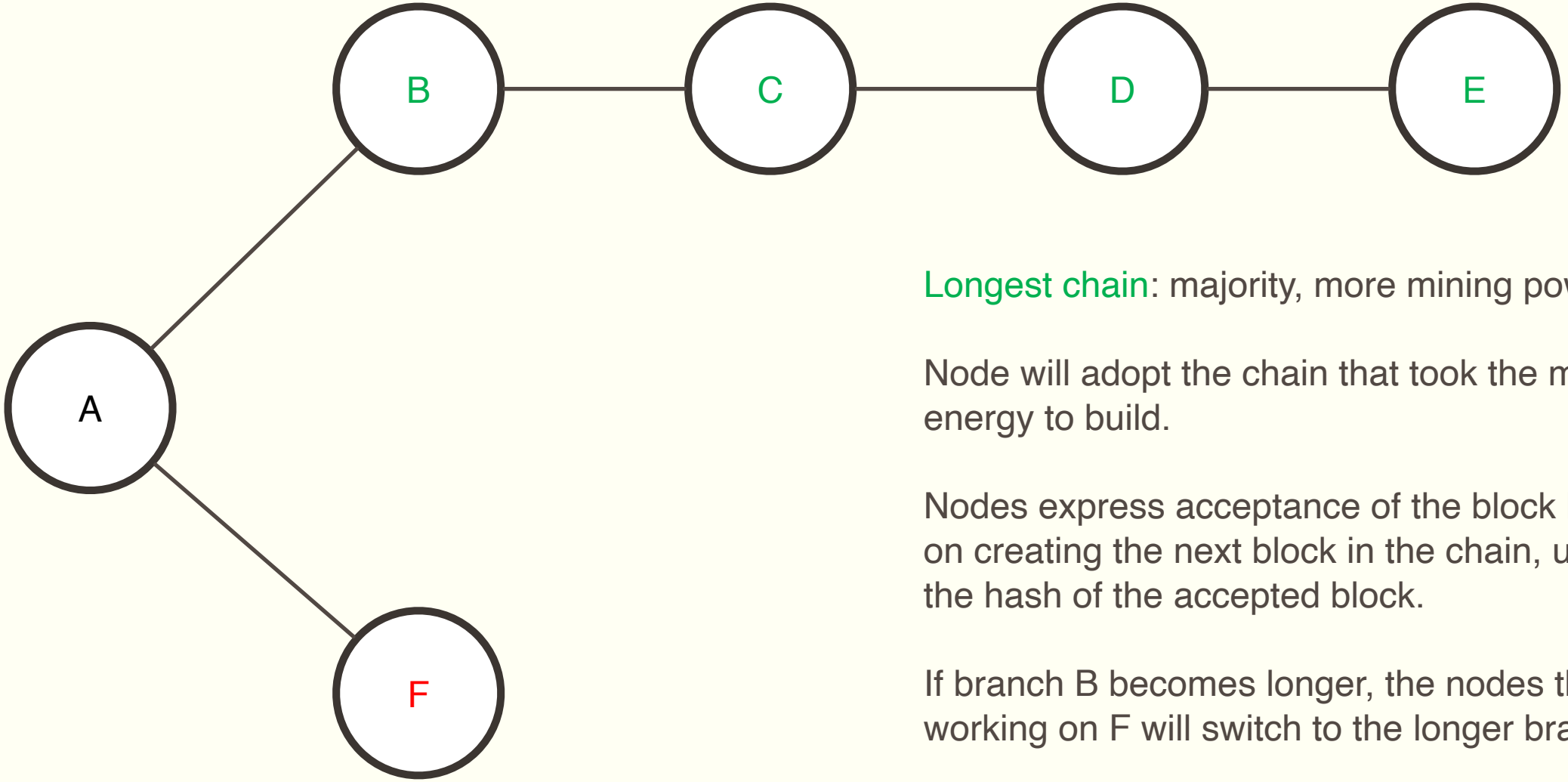


Fork: Two blocks are simultaneously added with the same previous block hash.

Bitcoin: block time 10min, Ethereum 15sec

Fork probability higher in Ethereum.

Fork: dispute correct order of valid transactions.

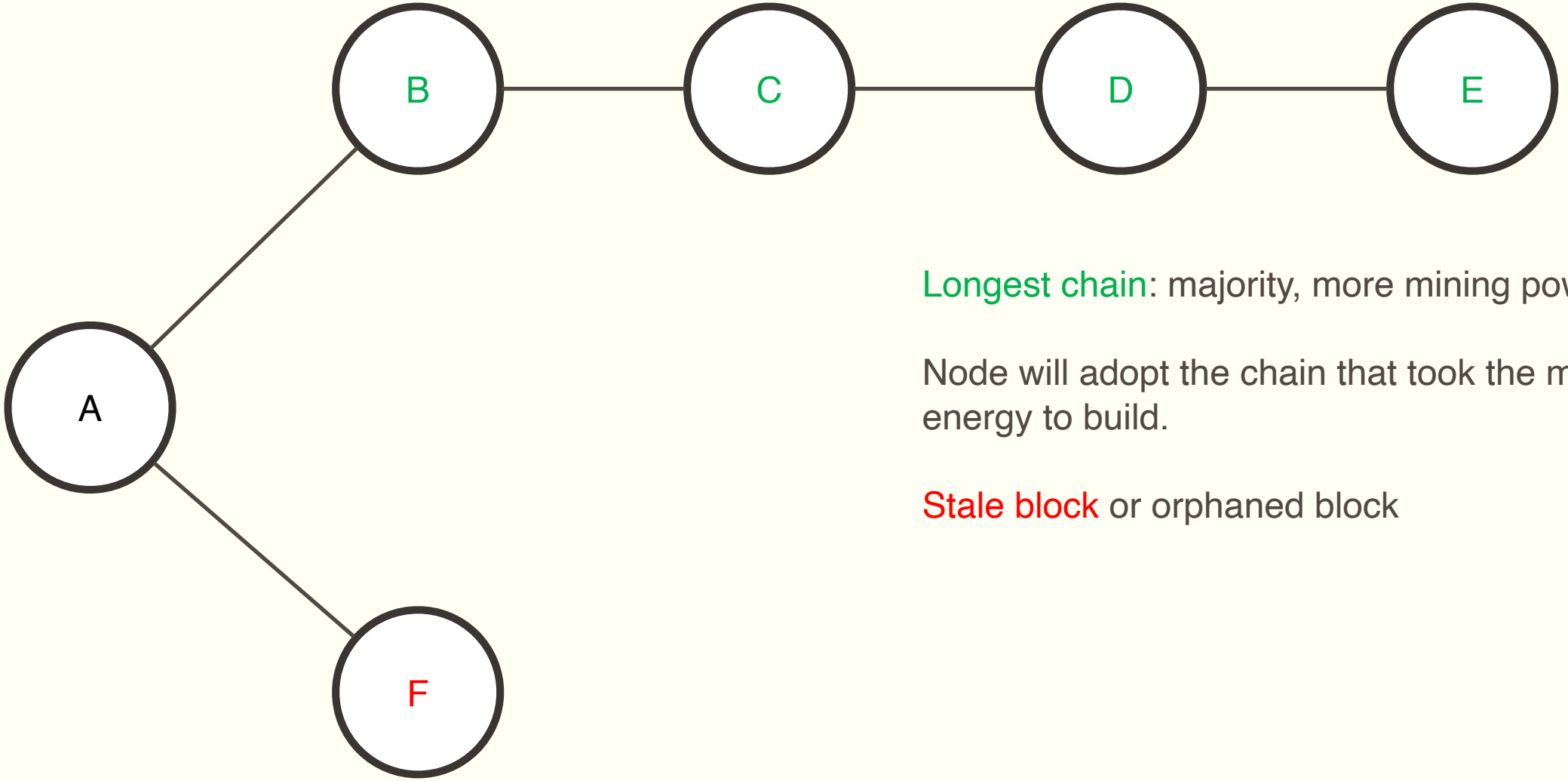


Longest chain: majority, more mining power

Node will adopt the chain that took the most energy to build.

Nodes express acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block.

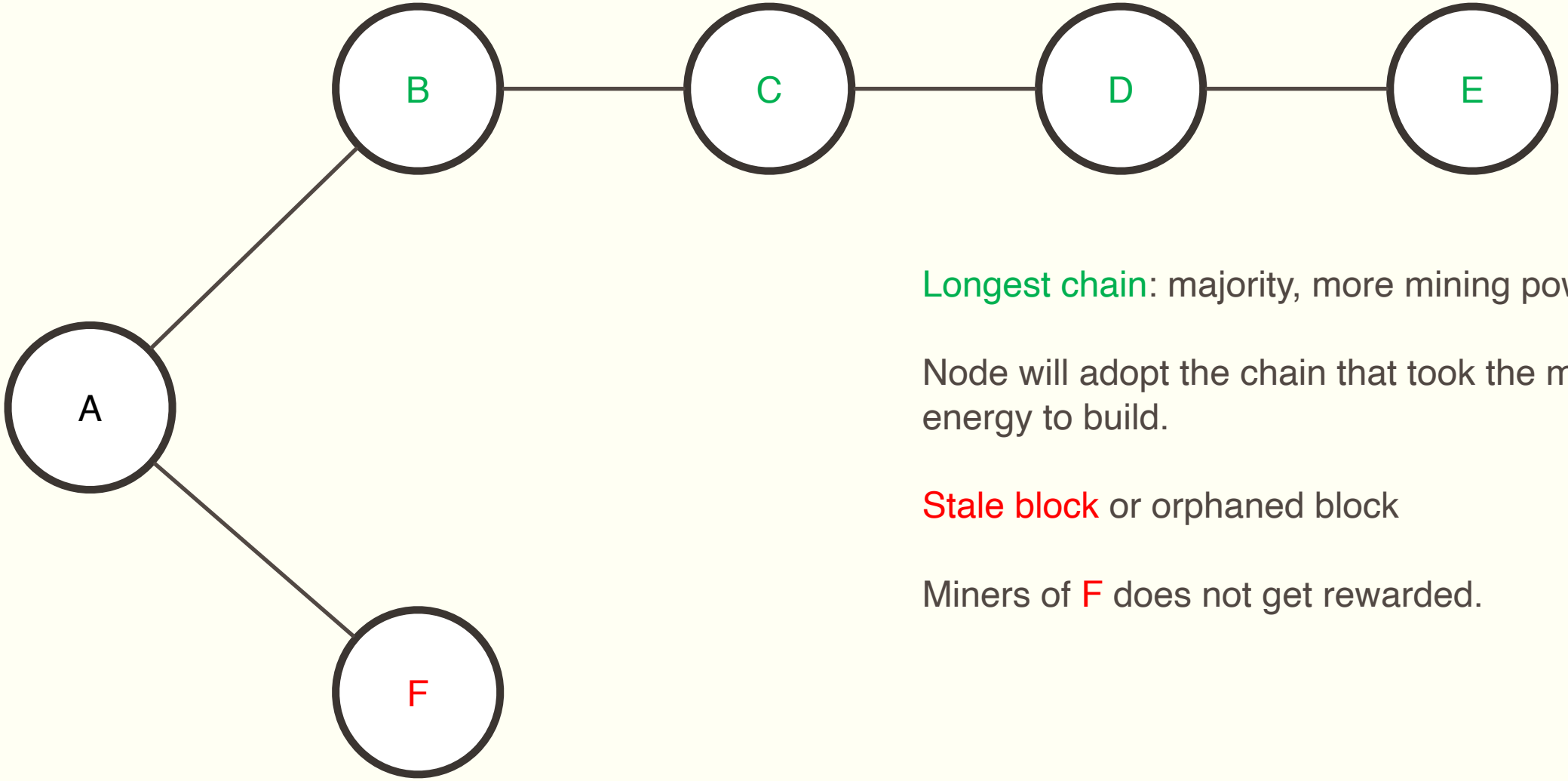
If branch B becomes longer, the nodes that were working on F will switch to the longer branch.



Longest chain: majority, more mining power

Node will adopt the chain that took the most energy to build.

Stale block or orphaned block

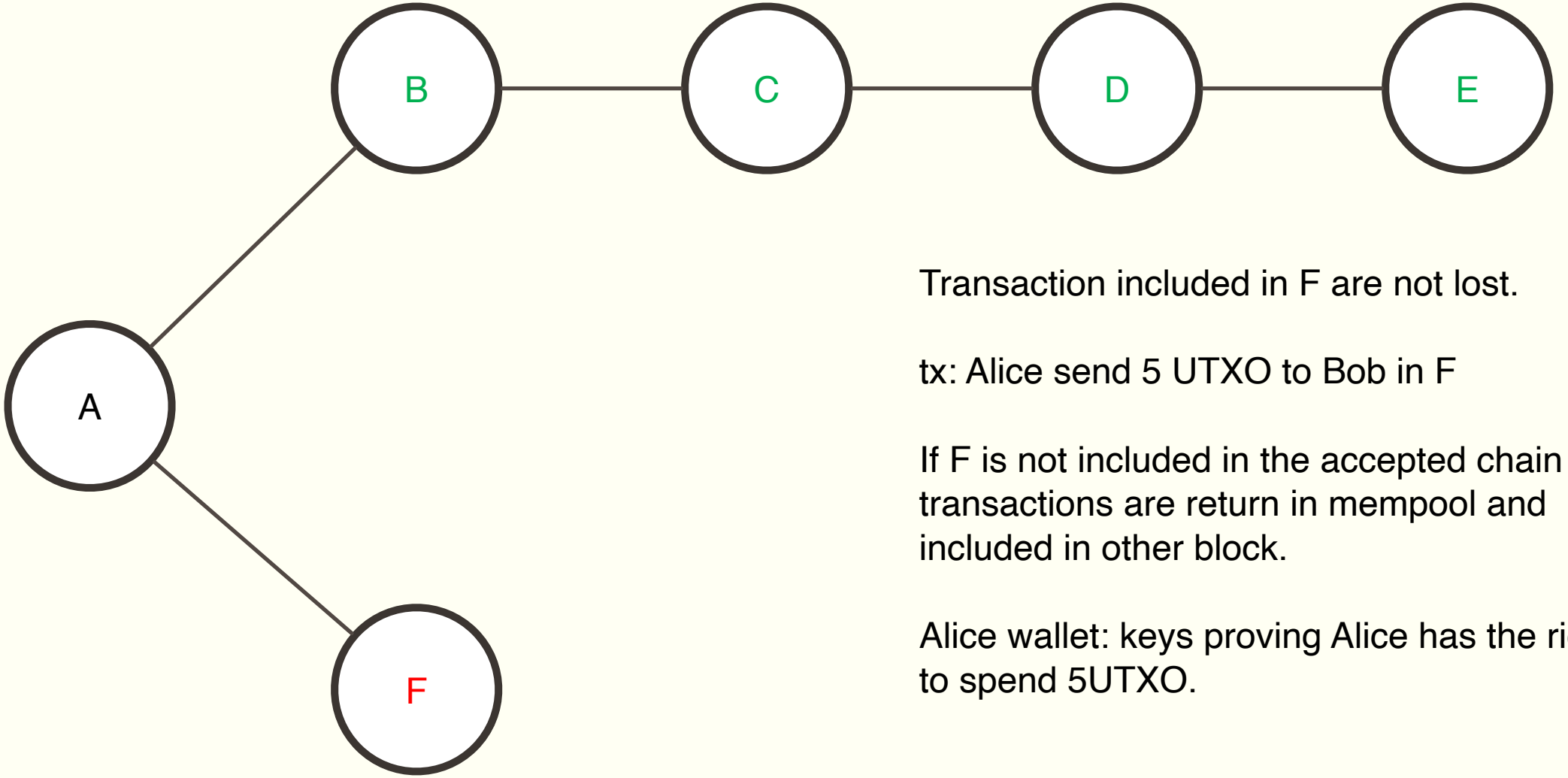


Longest chain: majority, more mining power

Node will adopt the chain that took the most energy to build.

Stale block or orphaned block

Miners of F does not get rewarded.

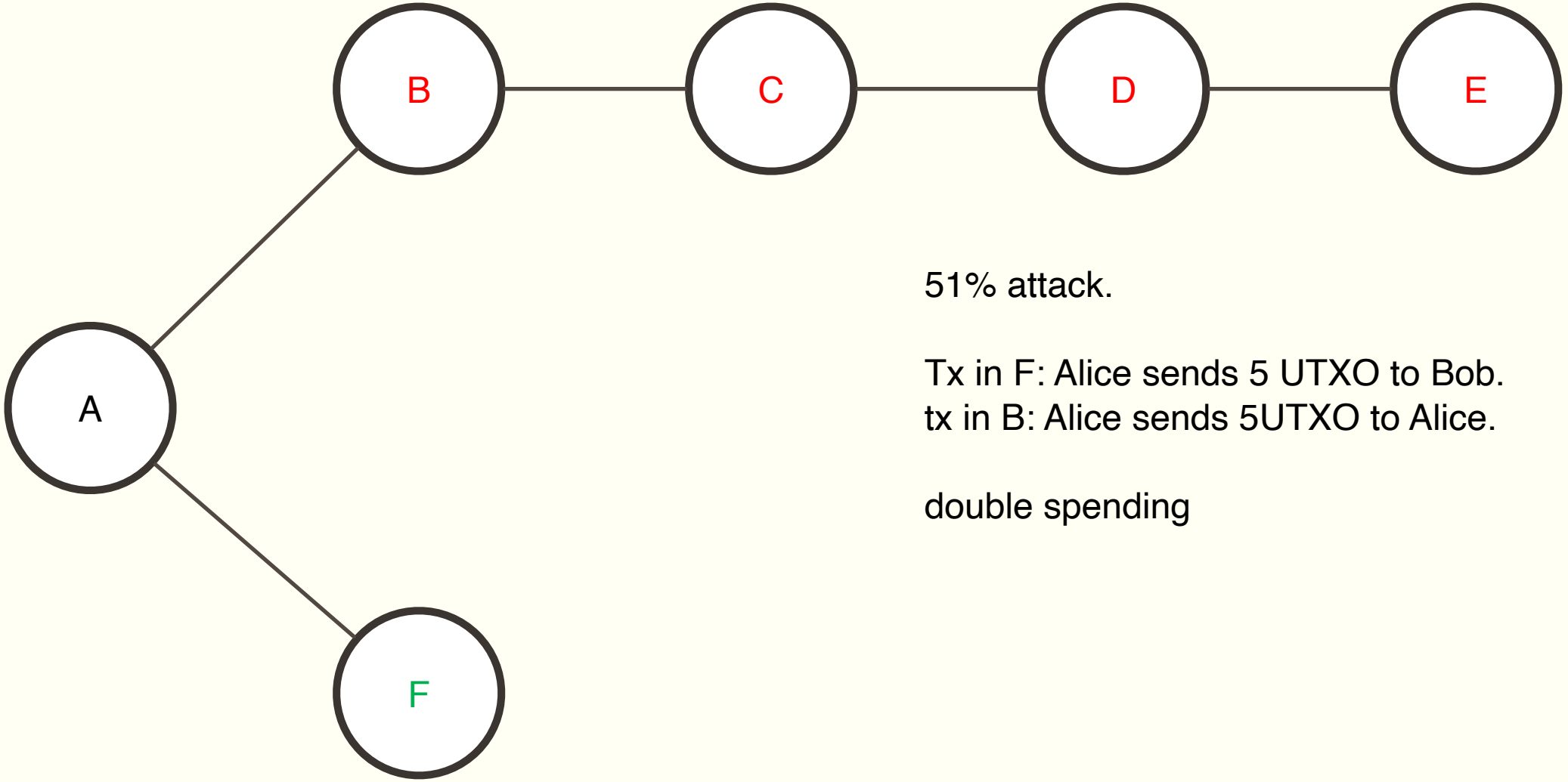


Transaction included in F are not lost.

tx: Alice send 5 UTXO to Bob in F

If F is not included in the accepted chain transactions are return in mempool and included in other block.

Alice wallet: keys proving Alice has the right to spend 5UTXO.



51% attack.

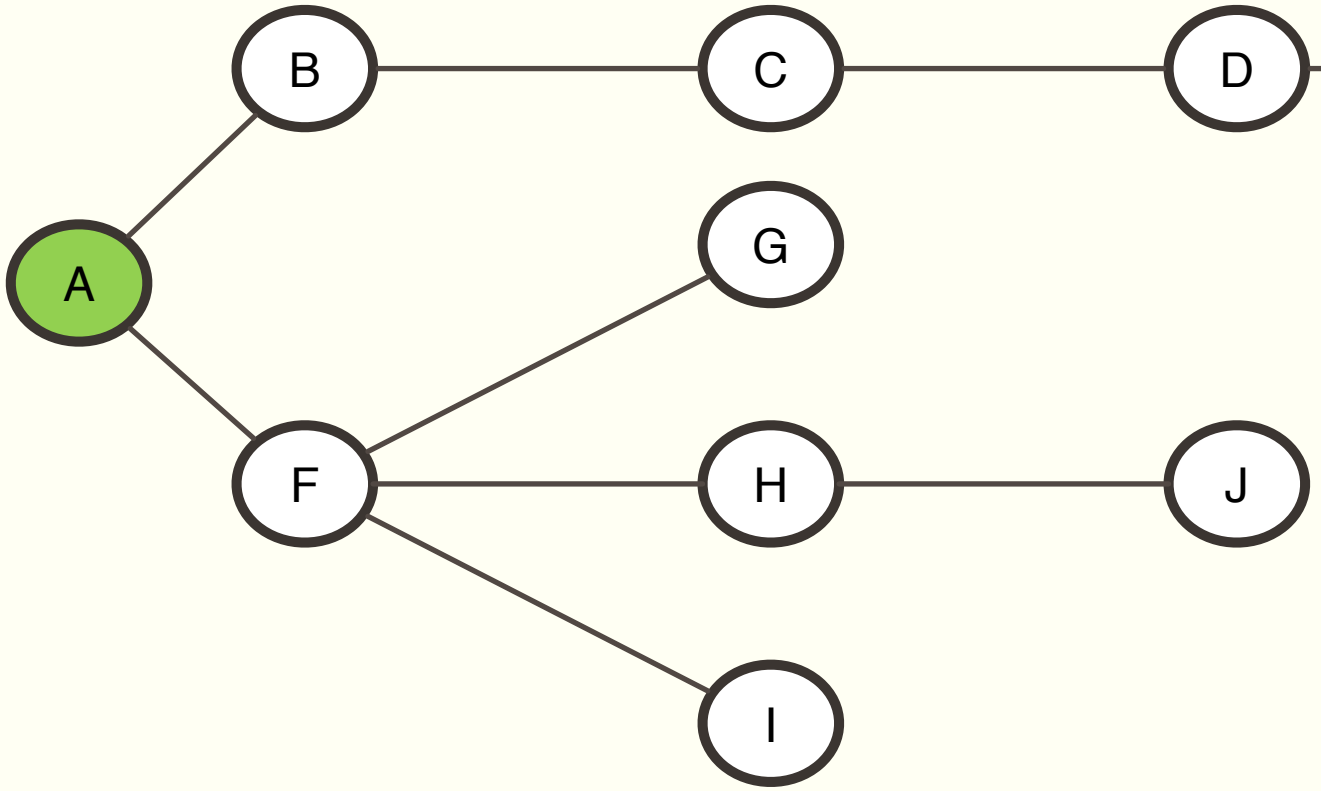
Tx in F: Alice sends 5 UTXO to Bob.
tx in B: Alice sends 5UTXO to Alice.

double spending

GHOST PROTOCOL

GHOST – Greedy Heaviest Observer SubTree

- To increase security miners that did not win are also part of total computing power.
- Power is divided between a larger set of nodes.
- GHOST partially rewards orphaned blocks (uncles) and incorporates them into the main chain.
- GHOST chose the branch with the highest cumulated difficulty.
- Step to the creation of PoS
  - The creator of a new block is chosen from a pool of users that stake assets.
  - 51% attack possible if attackers own 51% of total assets.
  - Rewards and penalties motivates good behavior.



nr_desc(B)

if children (B) = ∅ return 1

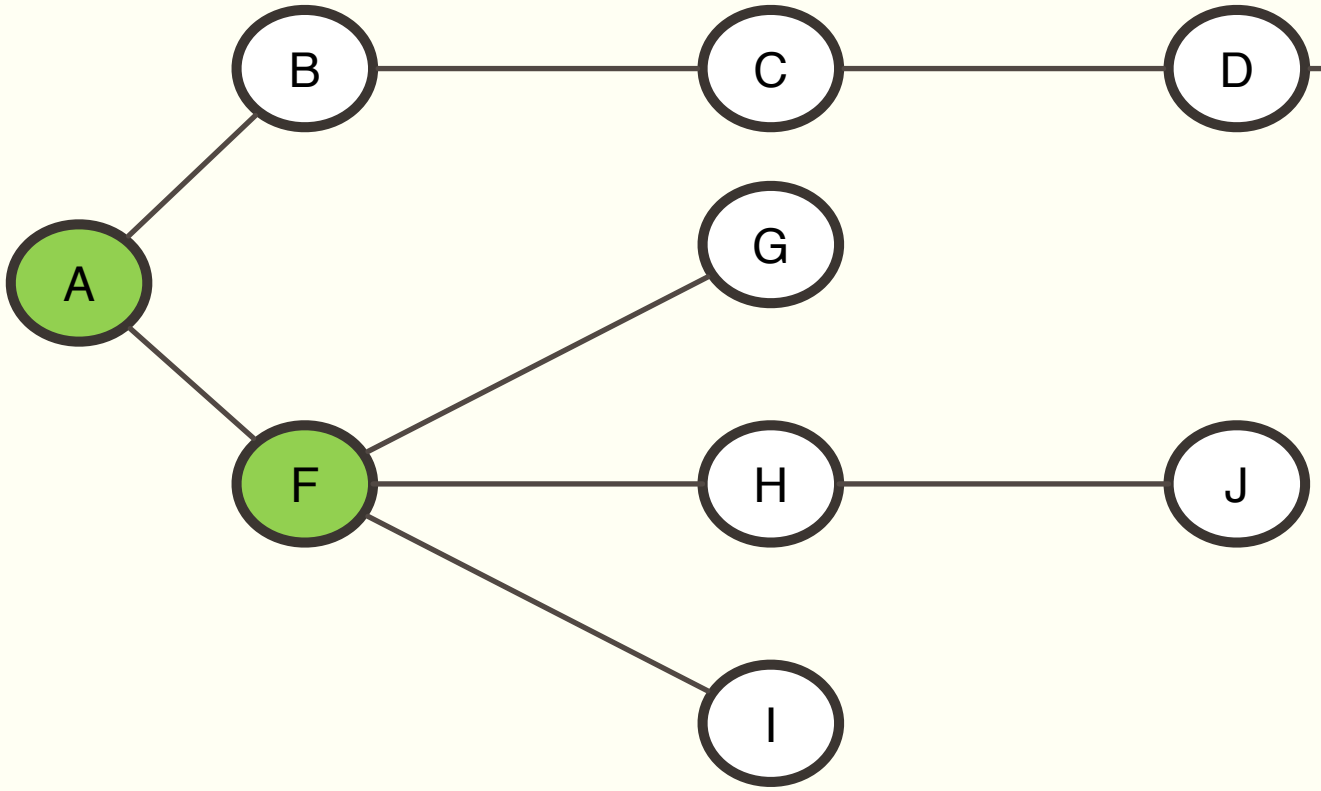
else return 1 + $\sum_{C \in children(B)}$ nr_desc(C)

B = genesis_block

CH=B

while (children(B) <> ∅)

B = $\underset{C \in children(B)}{\text{argmax}}$ nr_desc(C)



nr_desc(B)

if children (B) = ∅ return 1

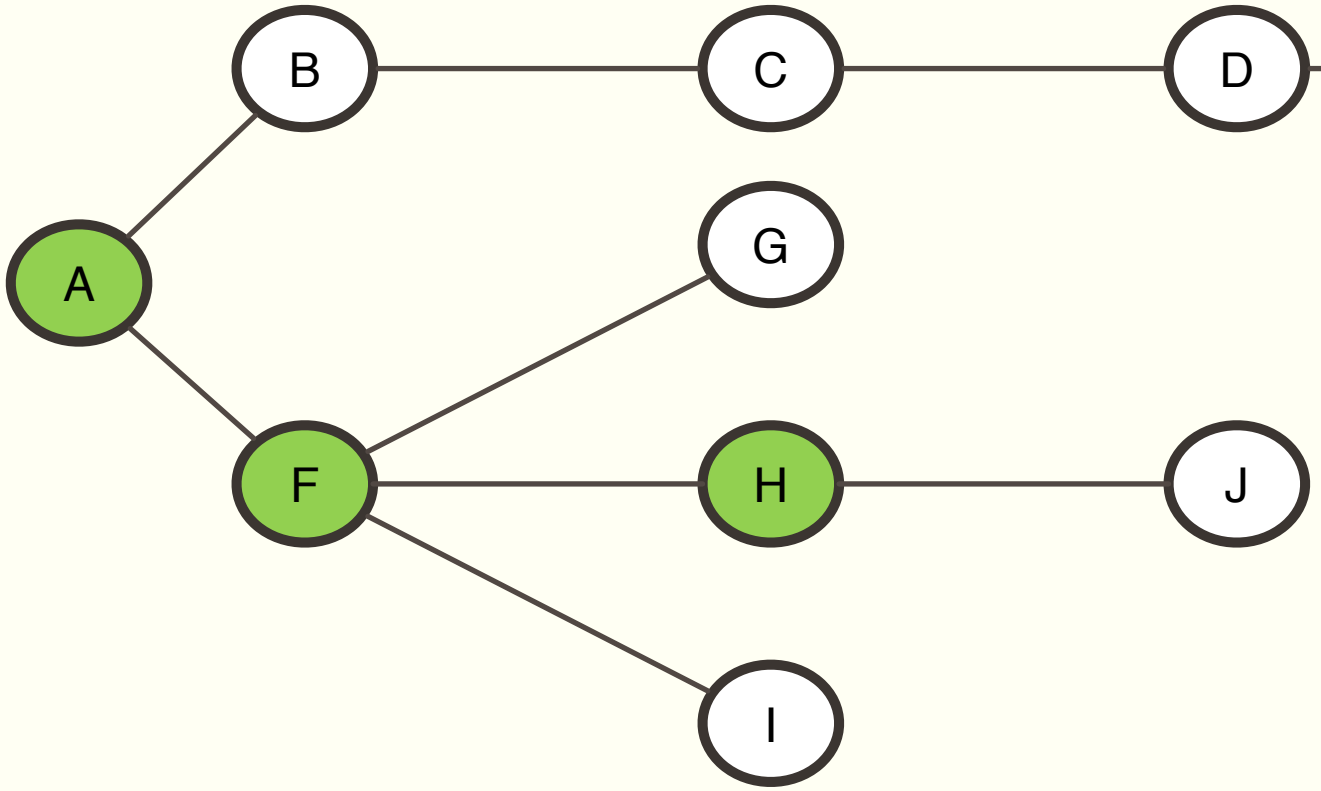
else return 1 + $\sum_{C \in children(B)}$ nr_desc(C)

B = genesis_block

CH=B

while (children(B) <> ∅)

B = $\underset{C \in children(B)}{\text{argmax}}$ nr_desc(C)



nr_desc(B)

if children (B) = ∅ return 1

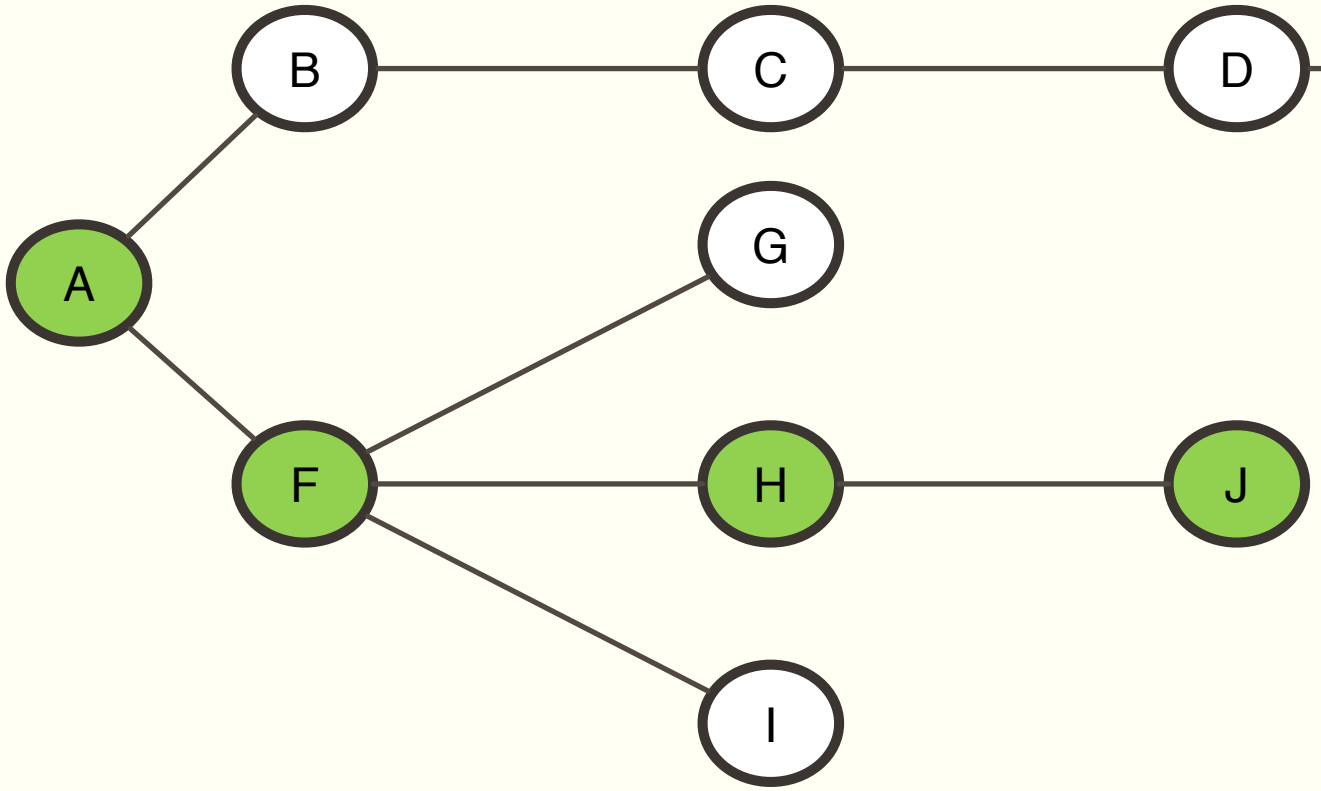
else return 1 + $\sum_{C \in children(B)}$ nr_desc(C)

B = genesis_block

CH=B

while (children(B) <> ∅)

B = $\underset{C \in children(B)}{\text{argmax}}$ nr_desc(C)



nr_desc(B)

if children (B) = ∅ return 1

else return 1 + $\sum_{C \in children(B)}$ nr_desc(C)

B = genesis_block

CH=B

while (children(B) <> ∅)

B = $\underset{C \in children(B)}{\text{argmax}}$ nr_desc(C)

GHOST – Greedy Heaviest Observer SubTree

- In Ethereum a modified ghost protocol is used.
- Transaction fees are not awarded to uncles but a stale block receives a reward of 87.5% of base reward.
- A block must specify a parent and 0 or more uncles.
- An uncle included in the block B:
  - Direct child of the k-th generation ancestor of B, 2<=k<=7.
  - It cannot by an ancestor of B.
  - Must have a valid block header.
  - Must differ from all uncles included for block B.
  - For every uncle U in block B, the miner of B gets an additional 3.125%