Three Dice Decentralized Consensus Algorithm

Step 1: - Independent verification of each transaction

Transactions creation and verification process:

1. Collecting UTXO
    - Bitcoin full nodes track all available and spendable outputs, known as unspent transaction outputs, or UTXO.
2. Providing the appropriate unlocking scripts
3. Constructing new outputs assigned to a new owner
4. Every bitcoin node that receives a transaction will verify the transaction.

Step 2: - Independent aggregation of transaction into candidate blocks

After validating transactions, a bitcoin node will add them to the memory pool, or transaction pool, where transactions await until they can be included (mined) into a block.

Step 3: - Independent verification of each block

Process done by every node

1. The node receives newly solved blocks sent from the miners.
2. The node validates the newly solved blocks.
    - The block data structure is syntactically valid
    - The block header hash is less than the target (enforces the Proof-of-Work)
    - The block timestamp is less than two hours in the future (allowing for time errors)
    - The block size is within acceptable limits
    - The first transaction (and only the first) is a Coinbase transaction
        - A dishonest miner could write themselves a Coinbase transaction for a thousand bitcoin instead of the correct reward.
        - An invalid Coinbase transaction would make the entire block invalid.
    - All transactions within the block are independently verified.
3. The validated blocks are added to the blockchain.
    - The honest miners of the solved blocks can spend their earned rewards.
    - The dishonest miners will have their blocks rejected and
        - lose the reward
        - waste the effort expended to find a Proof-of-Work solution, thus incurring the cost of electricity without compensation.
4. The node propagates the valid blocks.

Step 4: - Independent selection of blockchain

- The final step in bitcoin's decentralized consensus mechanism is
    - the assembly of blocks into chains
    - the selection of the chain with the most Proof-of-Work.
- Process of assembly a new block into chains

1. When a new block arrives, it will be verified every node.
2. The node will try to slot a new block that has been verified into the existing blockchain.
3. The node will look at the block's "previous block hash" field, which is the reference to the block's parent.
4. The node will attempt to find that parent in the existing blockchain.
5. Most of the time, the parent will be the "tip" of the main chain, meaning this new block extends the main chain.
    - For example, the new block 277,316 has a reference to the hash of its parent block 277,315.
    - Most nodes that receive 277,316 will already have block 277,315 as the tip of their main chain and will therefore link the new block and extend that chain.
6. Secondary chain branches the main chain as a result of an almost simultaneous mining of blocks at the same height.
7. If the block cannot find a parent (mostly because of child arriving before parent), it becomes an orphan and will join the main chain after the parent joins.

- Only the new blocks satisfying validation criteria are maintained by every node:

A. **Main Blockchain**: Those connected to the main blockchain

- The "main chain" at any time is whichever valid chain of blocks has the most cumulative Proof-of-Work associated with it.

- Usually this is also the chain with the most blocks in it, unless there are two equal-length chains and one has more Proof-of-Work.

- Consensus among
  - All nodes
    - By selecting the greatest-cumulative-work valid chain, all nodes eventually achieve network-wide consensus.
    - Temporary discrepancies between chains are resolved eventually as more Proof-of-Work is added, extending one of the possible chains.
  - Mining nodes
    - Mining nodes "vote" with their mining power by choosing which chain to extend by mining the next block.
  - When they mine a new block and extend the chain, the new block itself represents their vote.

B. **Secondary Blockchain**: Those that form branches off the main blockchain

- The main chain will also have branches with blocks that are "siblings" to the blocks on the main chain.
  - These blocks are valid but not part of the main chain. They are kept for future reference, in case one of those chains is extended to exceed the main chain in work.
  - Secondary chains occur as a result of an almost simultaneous mining of blocks at the same height.
  - Main chain is replaced by the secondary chain
    - Sometimes, the new block extends a chain that is not the main chain. In that case, the node will attach the new block to the secondary chain it extends and then compare the work of the secondary chain to the main chain.
  - If the secondary chain has more cumulative work than the main chain, the node will reconverge on the secondary chain, meaning it will select the secondary chain as its new main chain, making the old main chain a secondary chain.
    - If the node is a miner, it will now construct a block extending this new, longer, chain.

C. **Orphan Blocks**: Those that do not have a known parent in the known chains

- If a valid block is received and no parent is found in the existing chains, that block is considered an "orphan".
  - Orphan blocks are saved in the orphan block pool where they will stay until their parent is received.
    - Once the parent is received and linked into the existing chains, the orphan can be pulled out of the orphan pool and linked to the parent, making it part of a chain.
      - Orphan blocks usually occur when two blocks that were mined within a short time of each other are received in reverse order (child before parent).

Simple Target:

What is the probability of win if the target is 12?

Target is 12

- The player must throw 11 = 12 - 1 or less to win.
  - The player will only lose if he/she throws double-six.
- The probability of win is 35/36.

Difficult Target:

What is the probability of win if the target is 5?

Target is 5: The probability of the sum is less than 5.

- The player must throw 4 = 5 - 1 or less to win.
  - More than half the dice throws will exceed the target and therefore be invalid.
- The probability of win is 6/36.