**Consensus Mechanism For Bitcoin Proof of work -**

**Transaction Validation Rules-**

1. Check syntactic correctness
2. Make sure UTXOs are not empty
3. Size in bytes per transaction <= MAX\_BLOCK\_SIZE
4. Each transaction value, as well as the total, must be in legal money range(transaction value should never exceed total bitcoins in circulation)
5. Make sure none of the inputs have hash=0, n=-1 (*coinbase* transactions)
6. Reject if we already have matching transaction in the pool, or in a block in the main branch
7. For each input, if the referenced output exists in any other transaction in the pool, reject this transaction.
8. For each input, look in the main branch and the transaction pool to find the referenced output transaction. If the output transaction is missing for any input, this will be an orphan transaction. Add to the orphan transactions, if a matching transaction is not in there already.
9. For each input, if the referenced output does not exist (e.g. never existed or has already been spent), reject this transaction
10. Reject if transaction fee (defined as sum of input values minus sum of output values) would be too low to get into an empty block
11. Verify the [scriptPubKey](https://en.bitcoin.it/wiki/Script) accepts for each input; reject if any are bad
12. Add transactions to transaction pool(mempool)
13. Relay transaction to peers

**Block Validation Rules-**

1. Check syntactic correctness
2. Reject if we have a duplicate block
3. Transaction list must be non-empty
4. Block hash must satisfy claimed *nBits(number of leading zeros)* proof of work
5. Block timestamp must not be more than two hours in the future
6. First transaction must be coinbase (i.e. only 1 input, with hash=0, n=-1), the rest must not be
7. For each transaction, apply steps 2,3 and 4 mentioned above
8. Verify Merkle hash for the block
9. Check if the prev block (matching *prev* hash) is in the main branch or side branches. If not, add this to orphan blocks, then query peer we got this from for 1st missing orphan block in *prev* chain; done with block
10. Check that *nBits* value matches the difficulty rules
11. For certain old blocks (i.e. on initial block download) check that hash matches known values
12. There are three cases to add new blocks: 1. block further extends the main branch; 2. block extends a side branch but does not add enough difficulty to make it become the new main branch; 3. block extends a side branch and makes it the new main branch.
13. Relay mined block output hashes to peers