Dean Makovsky
Cryptocurrency Cabal
Problem Set 2

1. There is nothing to prevent both double spending and greedy mining. I think the main assumption Satoshi makes is that the attacker has a small bitcoin value at the time of the attack, which means that the attacker gains relatively less from defrauding people when compared to mining new bitcoin. It also assumes that people will become aware of the greedy attacker and that a potential attacker would try this scheme more than once.
2. If the attacker has 45% of computation power, a payment recipient needs to wait for 340 blocks to be published after which the recipient has a 0.001 probability of losing the money.
3. I created a program around the code Satoshi provided, which output the following:
   p < 0.05

q=0.1	z=3	p=0.0131722
q=0.15	z=3	p=0.0442278
q=0.2	z=5	p=0.0274155
q=0.25	z=6	p=0.0499426
q=0.3	z=10	p=0.0416605
q=0.35	z=17	p=0.0450402
q=0.4	z=36	p=0.0487612
q=0.45	z=137	p=0.0493437

4.
   a) The cloud provider could store any one particular block and send it back to you as if it was any other block you stored. Since it was a valid signed block, you would think it is accurate when it is not.
   b) No, for the reason I just explained.
5.
   a) To perform any of these operations (read/write/verify), you must read the entire data base and compute the hash value of it yourself and ensure this value equals your locally stored value.

      To read a particular value, start reading the entire database and computing the running hash. When the desired data is found, store this locally and then finish the hashing for the rest of the table.

      To write a value, maintain two running hashes of the database. The first will be to validate the database and the second will be the new hash value after writing. Write the desired value at your leisure.

      The verify is the same as read, but don't store any data locally.
   b) It is linear (n) , since the entire database must be read each time.
6. (Still assuming that the block position is not included in the block, and assuming only the head of the Merkle tree is stored locally)
   a) Read: read the entire desired block, hash it, then get reconstruct all parent nodes of the Merkle tree and verify that the root is the same. This requires retrieving each sibling node down the desired branch from the database (or whatever holds the Merkle tree).

      Write: write the desired block, then reconstruct the Merkle tree nodes above that block. Again, this requires retrieving each sibling node of the desired branch.

      Verify (assuming the object is to verify the data, not just the Merkle tree values, though the answer wouldn't change): Linearly read all the data and recompute each hash value.

Also manually reconstruct the Merkle tree and compare these values to what is stored in the database.

b) Reading and writing is now logarithmic and verifying is still linear.

7.

a) There are 24 * 6 = 144 blocks per day, so this mining pool (alpha = 0.15) would expect to find 21.6 of them.

b) Expected number of blocks found: (αt)/10

8. I further assume that if there is a split blockchain, that each entire supernode acts as one unit and randomly picks a single chain off of which to continue instead of each entity in the supernode randomly picking for itself. By this assumption, the blockchain can never split into more than two chains.

   The probability that an orphan node is created is the probability of …

   Supernode1 finding a block and Supernode2 finding a second during the propagation

9. t

10. (b) Is gambling with bitcoin legal? (if online, or if in-person).
    What are the demographics of bitcoin criminals? How does he track them down? What if the criminals are outside US boundaries?