Joseph Tobin (checked over with Vignesh Kuppusamy)
Cryptocurrency Cabal

Problem Set 2

1. Satoshi is also assuming that bitcoin users will continue to trust the network and transact on the network once he gains the majority. Even if he plans on playing by the rules, if no one expects him to play by the rules, bitcoin networks can try to exit the currency.

2. If the the attacker is 340 blocks behind, and the attacker has a 45% chance of finding the next block, the probability of the attacker catching up is less than 0.1%.

3. $p < 0.05$

Q	Z
0.10	3
0.15	3
0.20	5
0.25	6
0.30	10
0.35	17
0.40	36
0.45	137

```
#include <math.h>
#include <iostream>

using namespace std;

int main() {


  double q;
  int z;
  /**/
   cout << "q: " << endl;
   cin >> q;

   while(cin.good()) {
   cout << "z: " << endl;
   cin >> z;

  double p = 1-q;
  double lambda = z * (q/p);
  double sum = 1.0;
  int i, k;
  for(k =0; k <= z; k++) {
    double poisson = exp(-lambda);
    for(i = 1; i <= k; i++) {
      poisson *= lambda / i;
    }
    sum -= poisson * (1-pow(q / p, z-k));
  }
  cout << sum << endl;
  }


  return 0;
}
```

I just used Satoshi's code and submitted values of Z until I got the first $p < 0.05$.

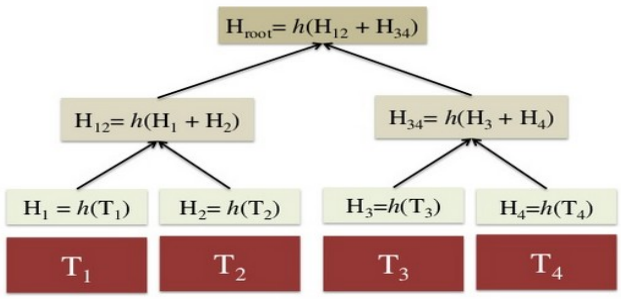
4a. If the bytes of data[i] don't indicate that is in fact from the I-th index, we cannot guarantee the server is storing the data in the correct places. It could be storing data[i] at position i+5, but we would not be able to tell.

4b. We cannot be sure that is indeed the data item we wrote, because if the signature if second preimage resistance is not guaranteed, the cloud server could replace the data entry with an entry that generates the same signature but that takes up significantly less space.

5a. To write the procedure would be to first add your entry to the database. Then, you would get all of the records in the database, concatenate every entry (including the one you just added), then hash that concatenation and store it locally.
To read, you would just get the record from the database.
To verify, you will have get every entry in the database, hash the concatenation, and then compare it to the locally stored hash.

5b. Reading is in constant time, because you just have to retrieve the single entry. (but if you are verifying after reading, it is linear because you have to get every entry). Writing is scales linearly with n because you have to get every entry in the database in order to create the new locally stored hash.

6a. To write the procedure would require you to insert the entry into the database. Then, you would form a Merkle Tree Hash. See below.



To read, you would just get the record from the database.
To verify, you will have to get the Merkle Tree hash of the database and compare it with your hash.

6b. To write, it will take log(n) time because it will take log(n) time to generate the hash that is stored locally. Reading is constant without verification and logarithmic with verification.

7. Given a, the proportion of hashing power a node possesses, Let v, the expected amount of blocks to be mined, if the node is mining selfishly, to be v =

$$\alpha^2 \times \left(2 + \frac{\alpha}{1 - 2\alpha}\right)$$

7a. Assuming (1 block / 10 mins) * (6 10-minute-blocks/hour) * 24 hours/day = 144 blocks / day.
If a = .15, then v = 0.049. Then, the expected value per block is (v/(v+ (1-(α^2)))), where (1-α^(2)) is the expected value of others honestly mining. Therefore, the pool is expected to find (0.049/(0.049 + 0.9775)) = 0.048 blocks / cycle. 0.048 blocks/cycle * 144 blocks/day = 6.87 blocks expected per day.

7b. Given the time t and probability hashing power a, the expected number of blocks to be mined, E, is the (chance of mining a block every cycle) / (minutes per cycle) * (the number of minutes passed). Therefore, E = ((v/(v+(1+α^(2)))/10) * t.

8. Expected number of orphan blocks per day= α * (1-α) * L * (1/86400)
// 86400= 60 seconds/minutes * 60 minutes/ hour * 24 hours

9. Because the selfish mining node has a low latency connection without, it still gets all of the blocks it would have already mined without the latency, but, in addition, it now receives all of the blocks that are disputed between itself and the other node. Therefore, it will have an expected number of blocks of (v/(v+ (1-(α^2)))) + α * (1-α) * L * (1/86400). Where v is described above, and α * (1-α) * L * (1/86400) is the number of orphaned blocks per day or the number of blocks which are disputed between the super nodes.

10b. Are people executing transactions on The Silk Road and similar websites actually untracable? How does the FBI approach anonymous online marketplaces? Do bitcoin users have any reason to believe the US government or another government would launch a malicious attack on the bitcoin network?