Problem Set #4

Due 11:59 pm, Tuesday, December 8th

Problem 1 Hashing (20 pts)

The hash table has 13 slots, and integer keys are hashed into the table with the following hash function H

```
int H (int key)
{
        x = ( key + 5 ) * ( key - 3 );
        x = int( x / 7 ) + key;
        x = x % 13;
        return x;
}
```

(a) Fill in the final hash table with the following keys: 17, 22, 73, 56, 310, 100, 230, 12, 42, 18, 19, 24, 49.

Chaining:

Slot	0	1	2	3	4	5	6	7	8	9	10	11	12
Contents	49		18		22, 42			310, 12, 24	73, 100, 19	17	56	230	

Linear Probing:

Slot	0	1	2	3	4	5	6	7	8	9	10	11	12
Contents	12	19	18	24	22	42	49	310	73	17	56	100	230

```
>>> def h(key):
...     x = (key + 5) * (key - 3)
...     x = int(x/7) + key
...     x = x % 13
...     return x
...
>>> h(17)
9
>>> h(22)
4
>>> h(73)
8
>>> h(56)
10
>>> h(310)
7
>>> h(100)
8
>>> h(230)
11
>>> h(12)
7
>>> h(42)
4
>>> h(18)
2
>>> h(19)
8
>>> h(24)
7
>>> h(49)
0
```

(b) List one or two methods that can handle collision in hashing.

A couple ways to handle collisions are the following:

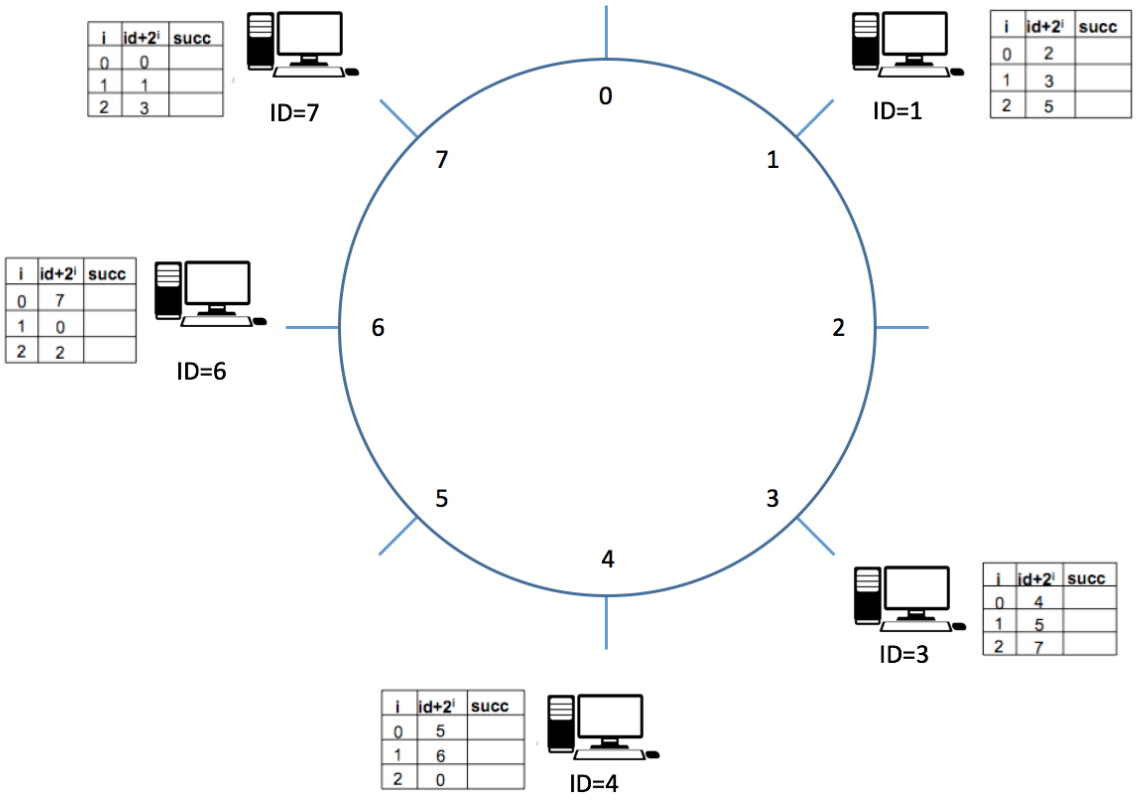
Chaining - just make it so the box is able to hold all things mapped to it (using linked-list or something similar)

Probing - the simplest would be linear probing where we just take the original hash and if the bucket is full we place it in the next available bin (by incrementing the hash for each full bin).

Problem 2 Distributed Hash Tables (20 pts)

There is a Chord DHT in Figure 1. with 5 nodes. The finger tables are listed beside the nodes. Each node may be storing some items according to the Chord rules (Chord assigns keys to nodes in the same way as consistent hashing)

Figure 1. Chord DHT for Problem 2



(a) Fill in the table for node id=1 and 7

	$ID + 2^i$	successor
0	2	3
1	3	3
2	5	6

Table for ID=1

	$ID + 2^i$	successor
0	0	1
1	1	1
2	3	3

Table for ID=7

(b) List the node(s) that will receive a query from node 1 for item 5 (item named by key 5)

Node one will query the key (5) and determine the successor to be node 6. The item is then passed on to node 6.

Only nodes 1 and 6 will see item 5.

Problem 3 Bloom Filters (10 pts)

Derive the probability of false positive rate after 10 keys (or elements) are inserted into a table of size 100. Assume that 5 hash functions are used to set up bit positions in the table for the keys (elements).

5 Hashing functions means: For each key (element) there will be 5 bits to be set to 1 in the table (it can also be less than 5 if the hash functions generate the same outputs).

Hint:
Assume **m** is the number of bits in the filter, **n** is the number of elements and k is the number of hash functions used.

After inserting one key, the probability of a particular bit being 0 is $(1 - \frac{1}{m})^{k}$. This is because k hash functions are independent, and each hash function will have $(1 - \frac{1}{m})$ probability for a particular bit remains 0. Then after inserting n keys, the probability for a particular bit remain 0 is $(1 - \frac{1}{m})^{kn}$
Now you have to apply this to the case of false positive.

So, given the information from the hint the chance of a bit being 0 after 10 keys have been inserted into a table of 100 with 5 hashes is, (1 - 1/100)^(5*10) = .605

This then means that there is a 1 - .605 = .395 chance that a bit will be on. For a false positive we would need to get 5 bits that are all on or .395^5 = .0096 or just about a 1% chance of a false positive.

Problem 4 P2P system (10 pts)

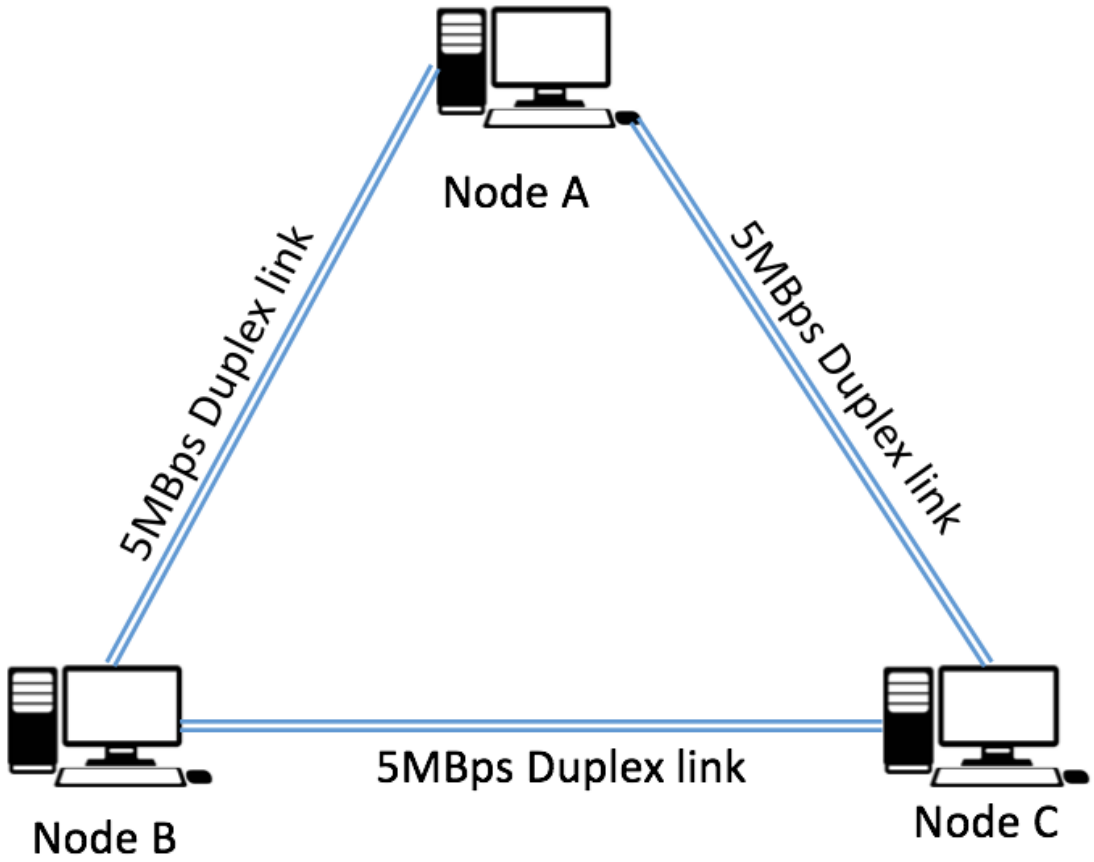


Figure 2. Network topology for Problem 4

3 nodes A, B and C are connected with each other via 5MBps duplex link as is now in Figure 2. Node A wants to share a 2500MB file to Node B and C. During the actual transmission 2.5MB piece of the file can be sent on the links each time. In the problem we ignore the RTT delay.

(a) What is the time of the sharing process using a centralized approach? (The process ends when B and C all received the file, and the centralized approach means B and C are communicating with A independently and there is no communication between B and C)

If A can put 2.5MB on the link to B and C each round then it will take 2500MB/ 2.5MB = 1000 rounds to complete the transmission.

(b) What is the ideal minimum time of the sharing process using the P2P approach? (P2P approach means after B and C received a piece from A, they immediately share the their piece to other party)

With the peer-to-peer approach B and C can share information already attained. A can send the start of the file to B and end to C and they can then exchange this information with one another. This would cut the transmission time in half. 500 rounds.

Problem 5 File Distribution (10 pts)

Consider distributing a file of F = 20 Gbits to N peers. The server has an upload rate of u_s = 30 Mbps, and each peer has a download rate of d_i = 2 Mbps and an upload rate of u. For N = 10 and 100 and u = 300 Kbps and 2 Mbps, prepare a chart giving the minimum distribution time for each combination of N and u for both client-server distribution and P2P distribution.

Client Server

min distribution =max(N*F/u_s, F/dmin)

U	N= 10	N= 100
300 Kbps	67000	67000
2 Mbps	10000	67000

Peer to Peer

U	N= 10	N= 100

300 Kbps	67000	67000
2 Mbps	10000	10000

```
>>> file = 20*10**9
>>> peers = 10
>>> uploadServer = 30*10**6
>>> dmin = 300*10**3
>>> max(peers*file/uploadServer, file/dmin)
66666.66666666667
>>> peers = 100
>>> max(peers*file/uploadServer, file/dmin)
66666.66666666667
>>> dmin = 2*10**6
>>> max(peers*file/uploadServer, file/dmin)
66666.66666666667
>>> peers = 10
>>> max(peers*file/uploadServer, file/dmin)
10000.0
>>> max(file/uploadServer, file/dmin, peers*file/(uploadServer + dmin * peers))
10000.0
>>> peers = 100
>>> max(file/uploadServer, file/dmin, peers*file/(uploadServer + dmin * peers))
10000.0
>>> dmin = 300*10**3
>>> max(file/uploadServer, file/dmin, peers*file/(uploadServer + dmin * peers))
66666.66666666667
>>> peers = 10
>>> max(file/uploadServer, file/dmin, peers*file/(uploadServer + dmin * peers))
66666.66666666667
>>>
```

Problem 6 BGP (20 pts)

1. Give the types of business relationships in BGP peering and mention who pays whom. What conditions make the Internet stable? Explain each condition in a line or two.

Customer-Provider: the provider charges the customer.
Peer-Peer: no charge.

Conditions:

1. Preference -> customer routes are chosen over peer/provider routes. This is done because it is choosing a revenue generating path rather than a costly or neutral path.

2. Export -> Only provide customer information to peers/routers. This limits the traffic that will be sourced through the node.

3. Topology -> The customer-provider graph is acyclic. Cycles defeat the point of the hierarchy and waste resources.

2. Please identify which of the following paths are valid, which of them are invalid based on the network topology of Figure 3.

Path 1 3 d invalid
Path 1 4 d valid
Path 8 d valid
Path 6 d valid
Path 4 d valid
Path 7 5 d valid
Path 7 5 3 d valid
Path 2 1 3 d invalid
Path 1 4 6 d valid

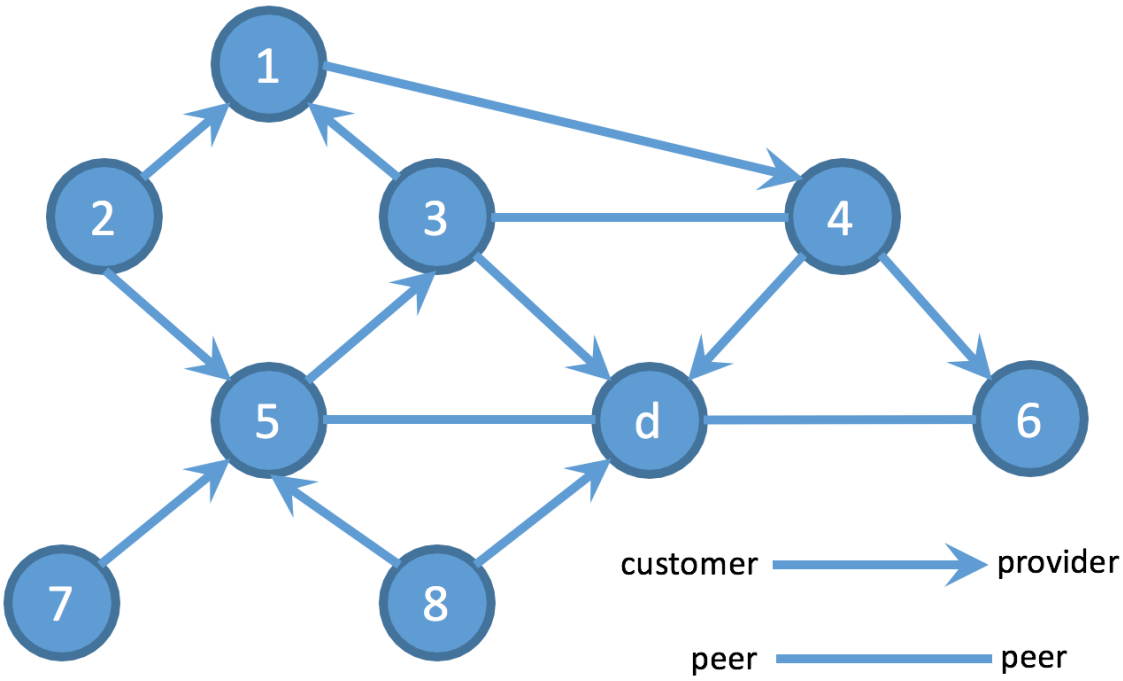


Figure 3. Network topology for Problem 6

Problem 7 Security (10 pts)

Diffie-Hellman Symmetric Key Exchange solves the key distribution issue of symmetric keys. Fill in the brackets below. Assume that Alice and Bob know p-ordered group G and a generator g, and Alice's and Bob's random seeds are ***a*** and ***b*** and their public keys are ***A*** and ***B***, respectively, and computes a shared key **s**. Please use ***p*** = 23, ***g*** = 11, ***a***= 13, and ***b*** = 8. Note that Eve is an eavesdropper and can see any communication between Alice and Bob.

Alice		**Bob**		**Eve**	
Known	Unknown	Known	Unknown	Known	Unknown
p = 23 ***g*** = 11 ***a*** = 13	***b***	***p*** = 23 ***g*** = 11 ***b*** = 8	***a***	***p*** = 23 ***g*** = 11	***a, b***
1. Alice chooses a private key *a* and sends its public key *A* to Bob					
A = [17]					
2. Bob chooses a private key *b* and sends its public key *B* to Alice					
		B = [8]			
3. Eve knows Alice's public key *A* and Bob's public key *B*					
A, B		***A, B***		***A, B***	
4. Alice and Bob calculate its shared key *s*					
s **= [** 18 **]**		***s*** = [18]			***s***