University of Nevada, Las Vegas Computer Science 477/677 Fall 2021

Answers to Assignment 5: Due Monday October 11, 2021 11:59 pm

1. Fill in the blanks.

   (a) In **perfect** hashing, no two data have the same hash value.

   (b) Using **Huffman** coding, the codons for the different symbols of a message may be written consecutively without spaces.

2. Write a recurrence for the time complexity of this function, then solve the recurrence.

```
void george(int n)
 {if(n > 0)
   {for(int i = 0; i < n; i++) cout << "hello" << endl;
    george(n/2); george(n/3); george(n/6);}}
```

   $T(n) = n + T(n/2) + T(n/3) + T(n/6)$

   $1/2 + 1/3 + 1/6 = 1$

   Therefore $T(n) = \Theta(n \log n)$ by the generalized master theorem.

3. A hash table with of size $m = 200$ is used to store 400 data items, using a pseudo-random hash function. The average number of items in a cell ("bucket") is clearly $\frac{400}{200} = 2$, but there could be empty cells. Approximately how many empty cells will there be, on the average? Pick the answer you think is closest.

   (a) 0

   (b) 5

   (c) 11

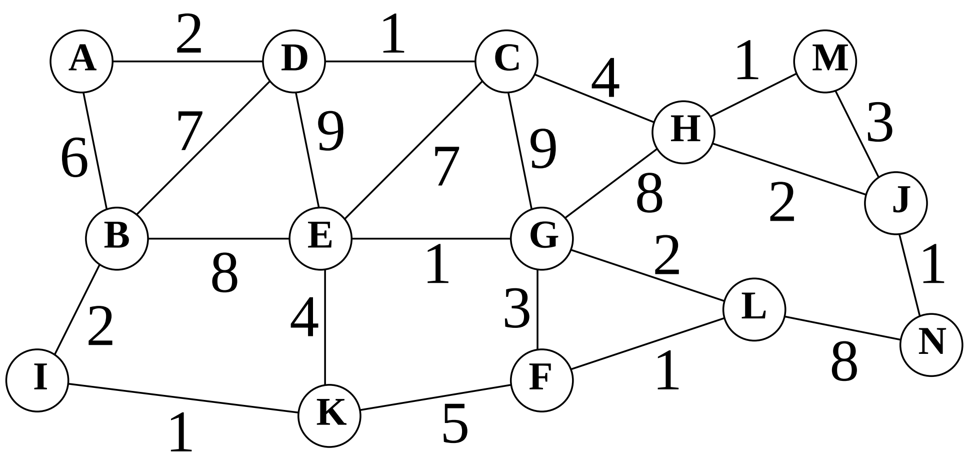
   (d) 27

   (e) 43

   (f) 84

   The expected proportion of empty cells is approximately $1/e^2$ which is approximately 0.13533... Since there are 200 cells, we expect that approximately 27 are empty.

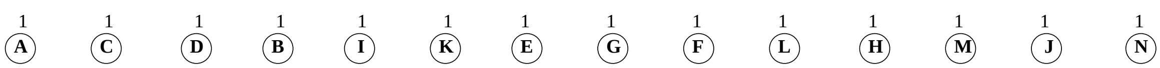
4. Walk through Kruskal's algorithm to find the minimum spanning tree of the weighted graph shown below. Show the evolution of the union/find structure at several intermediate steps. Whenever there is choice between two edges of equal weight, choose the edge which has the alphabetically largest vertex. Whenever there is a union of two trees of equal weight, choose the alphabetically larger root to be the root of the combined tree. Indicate path compression when it occurs.



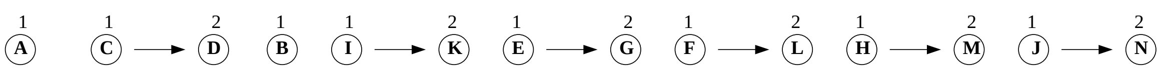
We first sort the edges by weight. The sorted list is:
DC, GE, KI, LF, MH, NJ, DA, IB, JH, LG, GF, MJ, HC, KE, KF, BA, ...
There are more edges, but I do not list them, since the algorithm halts before we get to them.

We process the edges in order of increasing weight, using the tie-breaker indicated in the description of the problem.

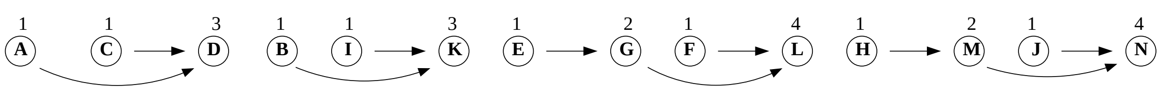
This is the initial union/find forest.



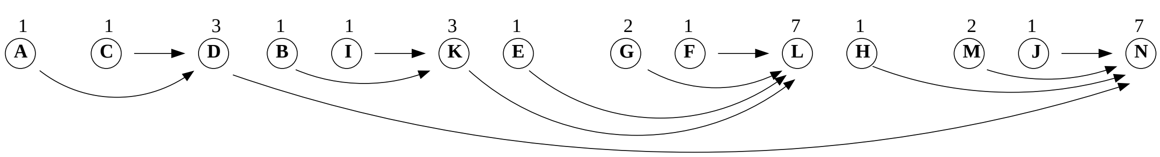
We now process edges of length 1, namely DC, GE, KI, LF, MH, and NJ.



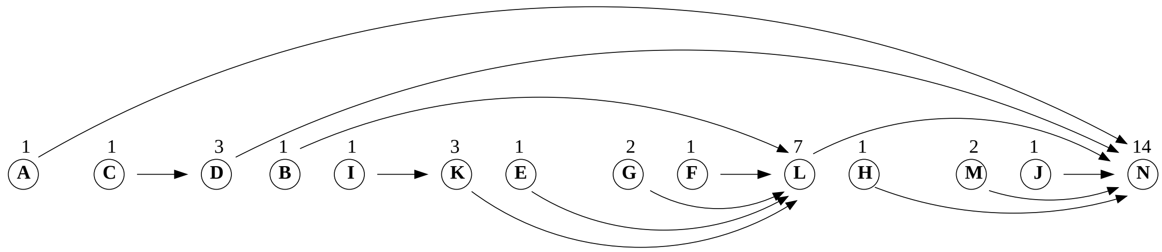
We now process edges of length 2, namely DA, IB, JH, and LG.



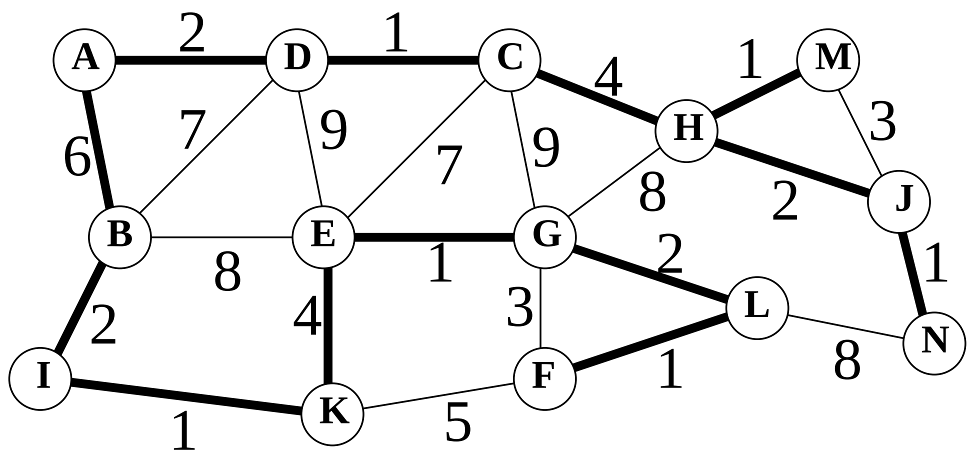
We now process edges of length 3, namely GF and MJ. There is no change in the data structure. We then process edges of length 4, namely HC and KE. There are two instances of path compression.



Finally, we process edge BA. There is one instance of path compression.

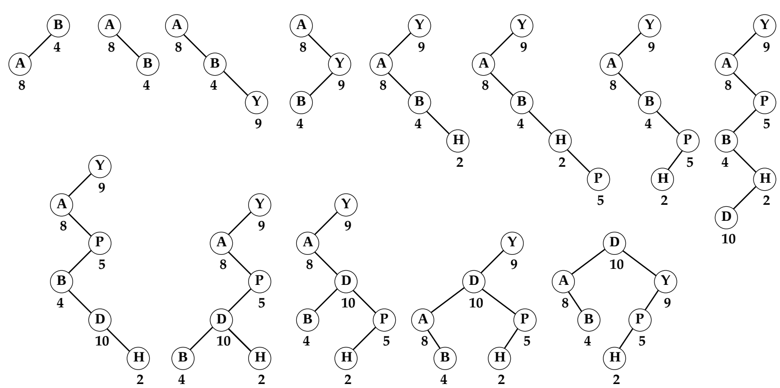


Since $n - 1 = 13$ edges have been selected, The minimal spanning tree is complete.



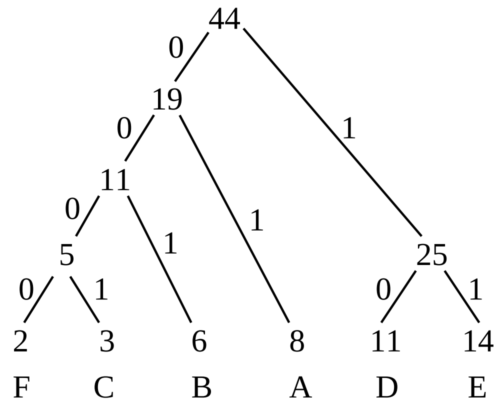
5. Insert the letters B, A, Y, H, P, D into an empty treap, where the "random" keys are given in the following table. Show the treap after each insertion and indicate all rotations.

A	8
B	4
D	10
H	2
P	5
Y	9



6. Find an optimal Huffman code on the alphabet A,B,C,D,E,F where frequencies are given in the following table.

A	8	01
B	6	001
C	3	0001
D	11	10
E	14	11
F	2	0000



7. A 3-dimensional $9 \times 7 \times 10$ rectangular array A is stored in main memory in row major order, and its base address is 8192. Each item of A takes one word of main memory, that is, one addressed location. Find the address of A[4][5][2].

   The offset is the number of predecessors, which is $4 \times 7 \times 10 + 5 \times 10 + 2 = 332$. The address is $8192 + 332 = 8324$.

8. You are trying to construct a cuckoo hash table of size 10, where each of the 9 names listed below has the two possible hash values, indicated in the array. Can you construct that table? Construct the table, or show that it can't be done by using Hall's marriage theorem.

	h1	h2
Ann	0	3
Bob	1	3
Ted	6	8
Sue	3	6
Gus	2	7
Cal	4	7
Dan	1	9
Sal	6	9
Eve	5	8

0	Ann
1	~~Bob~~ ~~Dan~~ Bob
2	Gus
3	~~Sue~~ ~~Bob~~ Sue
4	Sal
5	Eve
6	~~Ted~~ ~~Sue~~ Sal
7	
8	Ted
9	Dan

9. Give the asymptotic time complexity in terms of n, using Θ, O, or Ω, whichever is most appropriate.

   (a) $F(n) \geq 2F(\sqrt{n}) + 1$

   Substitute $m = \log n$, and $G(m) = F(n)$. We obtain:
   $G(m) \geq 2G(m/2) + 1$ $F(n) = G(m) \geq \Omega(m) = \Omega(\log n)$.

(b) $G(n) = 3(G(2n/3) + G(n/3)) + n^3$

Use the generalized master theorem.
$3(2/3)^3 + 3(1/3)^3 = 24/27 + 3/27 = 1$
$G(n) = \Theta(n^3 \log n)$