CS 477: Final Exam

Section: MW 2-3:50 pm

Total: 100pts @ 3 - 5pm on 05/04/2020

Instructions:

0. This is an open-book exam. You are free to consult the CLRS textbook. There are 9 problems in this exam. Please use your time wisely. Couple of them are from homework and couple of them are extensions from homework problems (that is simply running the algorithm doing the tracing).

1. I leave plenty of space on each page for your computation. If you need more sheet, please attach your work right behind the corresponding problem. If your answer is incorrect but you show the computation process, then partial credits will be given. Please staple your solution and use the space wisely.

2. Full credit will be given only to the correct solution which is described clearly. Convoluted and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

First Name:

Last Name:

Score: / **110**

Problem 1 Complexity Computation: 15pts (5+10)

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n-a) + T(a) + cn^2 + (n-a)$ where $a \geq 1$ and $c > 0$ are constants. For simplicty, you can assume that $n = ka$.
(a) Please draw the recursion tree to represent this formula

(b)Please show the complexity. [Note: $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$. You must show the computatin steps when summing up the cost of each step.]

Problem 2 Data Compression: 10 pts

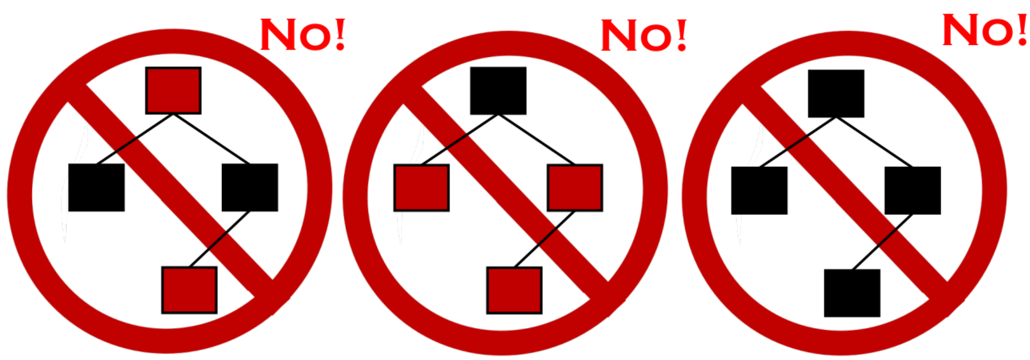
Consider the four variable-length codes shown as below: code 1 = (000, 100, 10, 11), code 2 =(0, 1, 00, 11), code 3 =(1, 0, 001, 0001), code 4 =(1, 01, 001, 000) for letters A, B, C and D. Which coding schemes are prefix-free?

Prefix-free code are:

Non-Prefix-free code are: **violation is:**

Problem 3 Red-Black Tree:10pts (3+3+4)

Rebalancing tree is important for binary-tree-like structures to avoid the skew problem. B-tree and RB-tree are two approaches that we covered this semester. In RB tree, we have the right rotation and the left rotation.
(a) Below are three trees (T1, T2, T3, left to right) that are not RB-trees. Why (explain for each)?

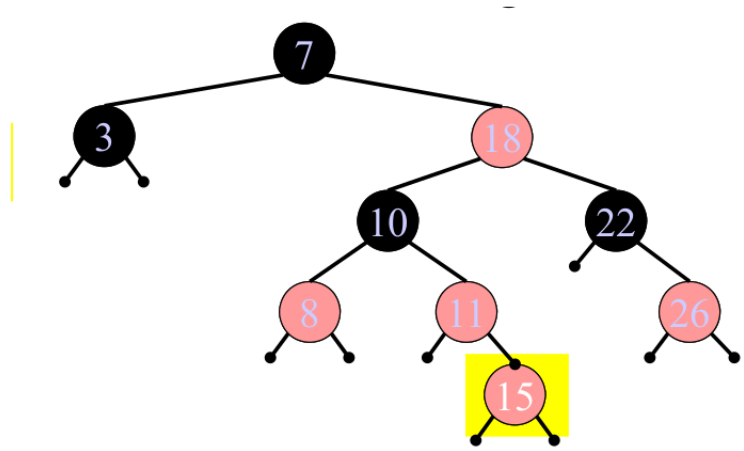


T1:

T2:

T3:

(b) We know that the height of a RB-tree is at most 2* OPT where OPT = $\log N$ with N being the total number of elements (nodes) in the data set. Given the following RB tree with 15 is the node to be inserted.



(1) What roation occurs between what two nodes when you are balancing the tree?

(2) Draw the resultant RB-tree.

Problem 4 Concept: BFS, DFS: 10pts

Given a balanced tree with branching factor = b and height = m. Please draw a simple graph (along with some brief description) for each sub question:
(a) BFS outperforms DFS in terms of time complexity

(b) DFS outperforms BFS int terms of time complexity

Problem 5 Heapsort: 5+5 points

What is your UID:

If UID is even, $\tau = 4$, else $\tau = 3$. A τ-nary heap is like a binary heap, but (with one possible exception) non-leaf nodes have τ children instead of 2 children. Here let us assume is min heap with n nodes. That is the parent is SMALLER than its children. Let us assume the root has index 0, and its children have index 1, 2, $\cdots$, d. And the same logic applies with the descendants.

(a) Assuming node at locaiton i also have τ children, what are the indices of its children?

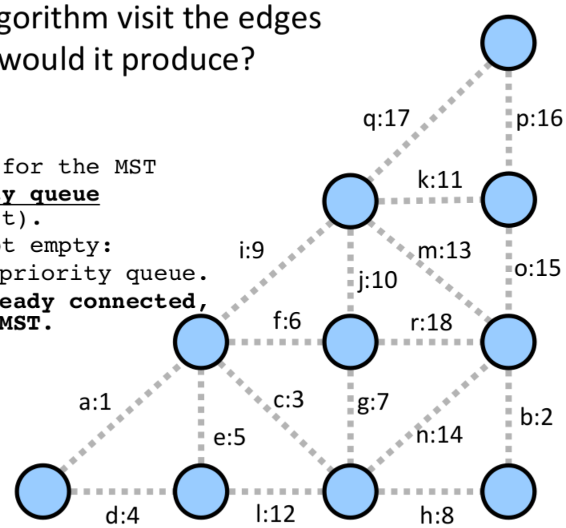
(b) Give an efficient pseudo code EXTRACT-MIN in a ternary min-heap. Analyze its running time in terms of n.

Problem 6 Minimal Spanning Tree (MST): Kruskal: 15pts

MST is to find a tree within a connected edge-weighted graph where the tree is supposed to be of minimal weight. Kruskal's algoroithm is one of the approaches.

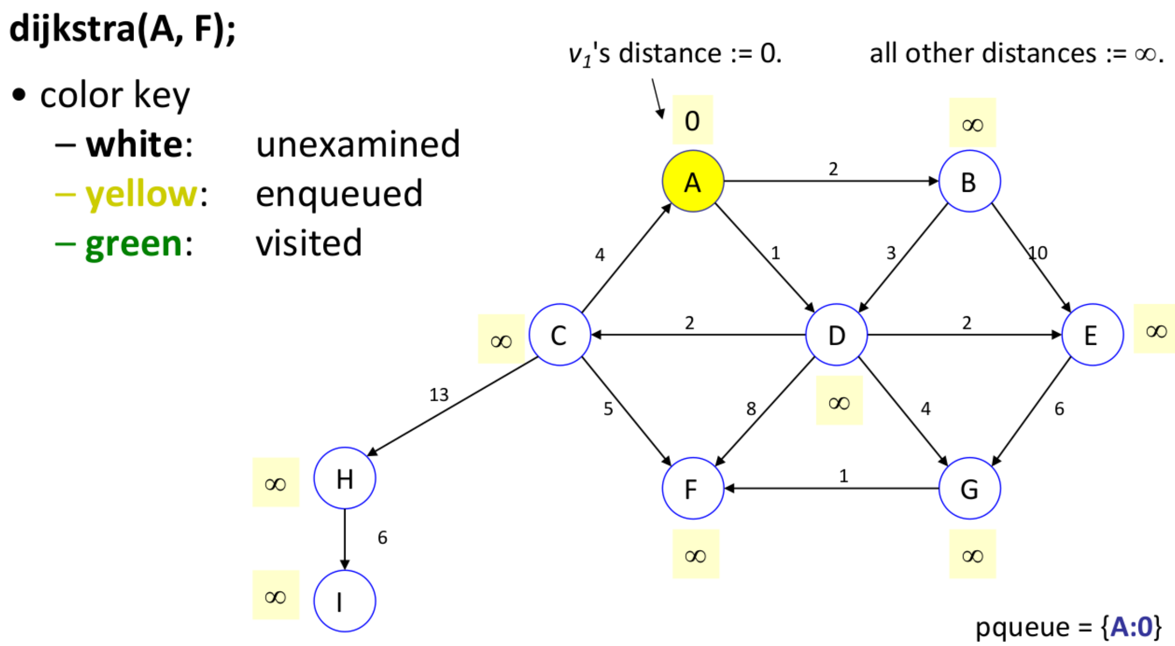
- In what order would Kruskal's algorithm visit the edges in the graph below? What MST would it produce?

```
function kruskal(graph):
  Start with an empty structure for the MST
  Place all edges into a priority queue
      based on their weight (cost).
  While the priority queue is not empty:
     Dequeue an edge e from the priority queue.
     If e's endpoints aren't already connected,
         add that edge into the MST.
     Otherwise, skip the edge.
```



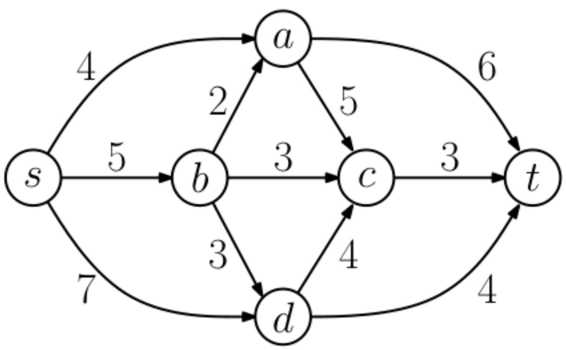
Problem 7 Shortest Path: Dijkstra's 15pts

Run Dijkstra's algorithm on the directed graph shown below from node A to node F. Please make sure you update the queue in every step and finally show the result.

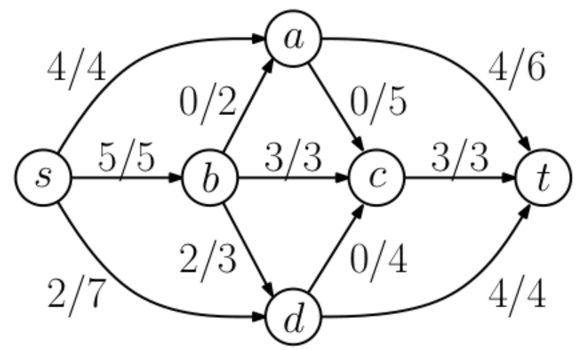


Problem 8 Maxflow MinCut: 15pts

We trace the partial execution of the Ford-Fulkerson algorithm on a sample network. (I) Consider the s-t network G shown below in figure (a), and consider the initial flow f in figure (b). Show the residual network Gf for this flow.



(a) Initial network G



(b): Initial flow f

(II)Find any s-t path in Gf . How much flow can you push along this path? Show the updated flow (in the same manner as Fig. b)

(III) Is this the final maximum flow in this network? (If not, keep running Ford-Fulkerson until you get the maximum flow, and show the final flow.) What is the value of the maximum flow?

Problem 9 String Matching: KMP: 10pts

Compute the prefix function π for the pattern p. In this case we have $p = abbabba$
We know that text $T = ababbabbabbababbabb$