CS251 - Homework 3: Trees and Graphs

Out: March 04, 2016 @ 9:00 pm
Due: March 25, 2016 @ 9:00 pm

Name: Dylan Mackey

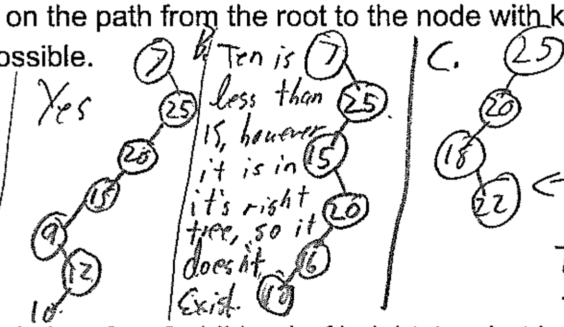
Read this before start: Not all questions are multiple choice, some of them may require an explanation or simulating an algorithm. In those questions give as much information you consider is required for the solution. Read carefully each question in order to identify what is being asked.

**** For all questions, any answer without an explanation (even though it is correct) will be graded with 0 points. ****

1 (10 pts). Consider binary search trees of size 30 containing keys 1 through 30. The query search(10) is executed. For each one of the following sequences explain whether there exists a tree so that the entries in the sequence are keys on the path from the root to the node with key 10. Either sketch a tree or explain why it is not possible.

✓ a) 7→25→20→15→9→12→10 ✓
~~b)~~ 7→25→15→20→16→10
~~c)~~ 25→20→18→22→16→10

A. Yes



B. Ten is less than 15, however it is in it's right tree, so it doesn't exist.

C. Since 22 is in 20's le tree, it has to be less.

2 (10 pts). For an arbitrary 2-3 tree (i.e., each node has 2 or 3 children) of height h, what is the minimum and maximum number of nodes such a 2-3 tree may have?

1 ✓ a) $2^{h+1}-1$, $\frac{3^{h+1}-1}{2} = 1$ ✓
11 ~~b)~~ $2^{h-1}-1$, $\frac{3^{h-1}-1}{2}$
8 ~~c)~~ 2^{h+1}, $\frac{3^{h+1}}{2}$
2 ~~d)~~ 2^{h-1}, $\frac{3^{h-1}}{2}$

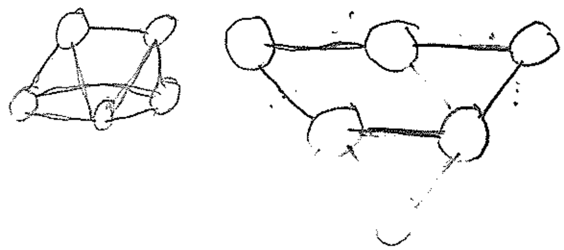
0

h = 2
max = 4
min = 3

Proof: 0	Proof: 1	Proof: 2
min: $2^1-1 = 1$	min: $2^2-1 = 3$	min: $2^3-1 = 7$
max: $\frac{3^1-1}{2} = 1$	max: $\frac{3^2-1}{2} = 4$	max: $\frac{3^3-1}{2} = 13$

3 (10 pts). A graph G is an undirected graph with n vertices and each vertex has exactly degree d (i.e., d edges touching it). Assume n is odd. What can you tell about d?

a) d must be an odd number
b) d must be an even number
c) d can be odd or even
d) None of the above



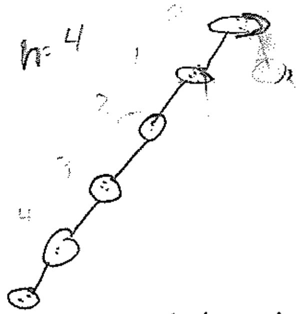
If d is odd, then one vertex will have d+1 edges.

4 (10 pts). We have n elements. How many binary trees of height *n-1* can be built from them?

* Check Piazza. Balanced??

a) 1
b) 2
c) $n!$
d) $2^{(n-1)}$

n=4



Only option is a completely unbalanced tree with the first node being height 0.

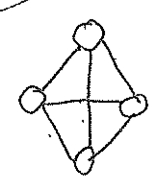
5 (10pts). Given a connected graph of n nodes and e edges, the minimum and maximum number of edges of the graph is:

max: $e \leq n(n-1)/2$
min: $n-1$

a) $n-1 \leq e \leq n(n-1)/2$
b) $n^2 \leq e$
c) $n \leq e \leq n^2$
d) None of the above

max: $\frac{4(4-1)}{2} = 6$, so $\frac{n(n-1)}{2}$

ex:

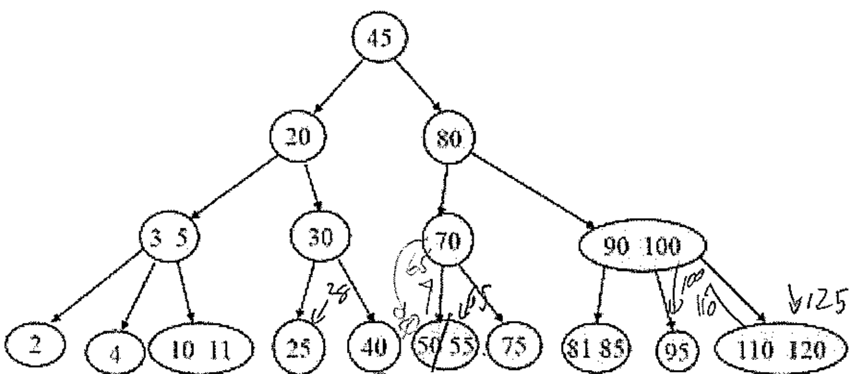


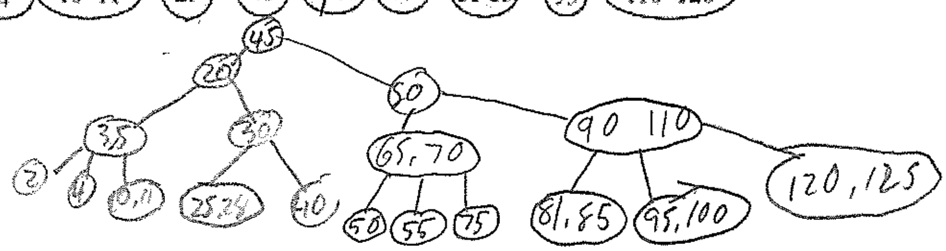
n=4
e=6

min:

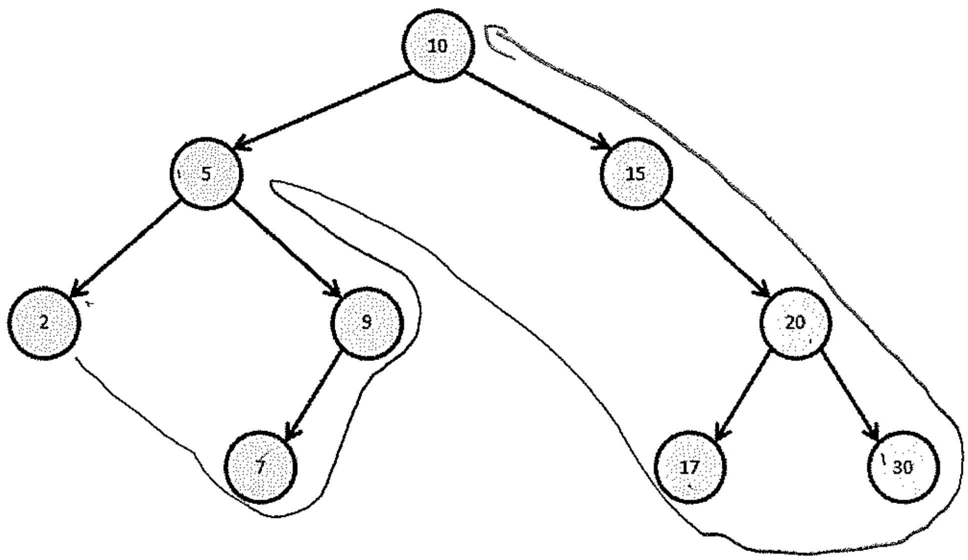
n=2, e=1, (n-1)

6 (10 pts). In the 2-3 tree given below, execute insert(28), insert(125) and insert(65) using analogous re-balancing operations as in a 2-3-4 tree but using a 2-3 tree (i.e., attempt at all times to keep the tree balanced). Draw the resultant 2-3 tree after executing the operations.





7 (10 pts). Given the tree below, which sequence is a post-order traversal?

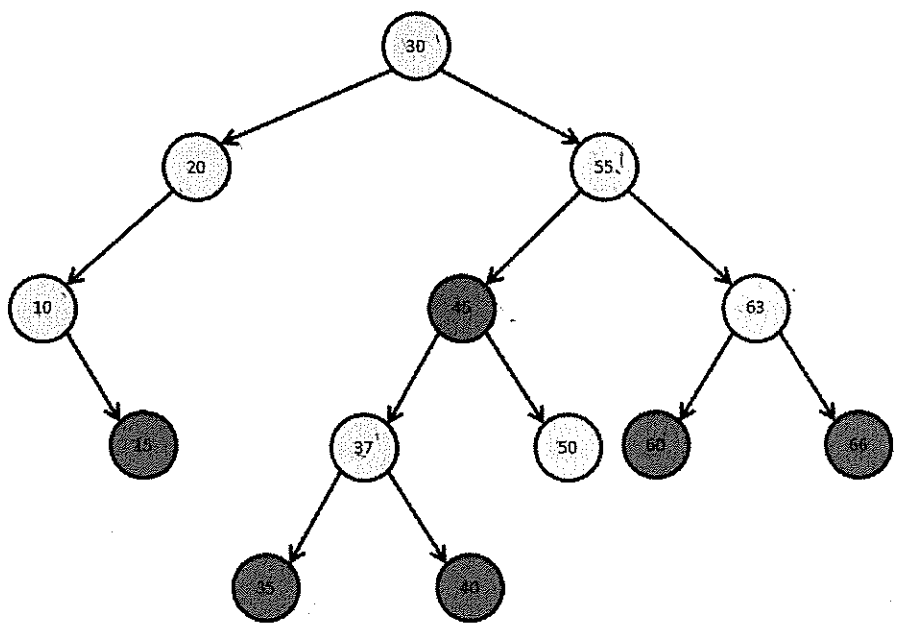


a) 10,5,15,2,9,20,7,17,30
b) 10,5,2,9,7,15,20,17,30
c) 2,5,7,9,10,15,17,20,30
d) 2,7,9,5,17,30,20,15,10

"Left, right, parent

8 (10 pts). Given the tree below, is it a valid Red-Black tree?

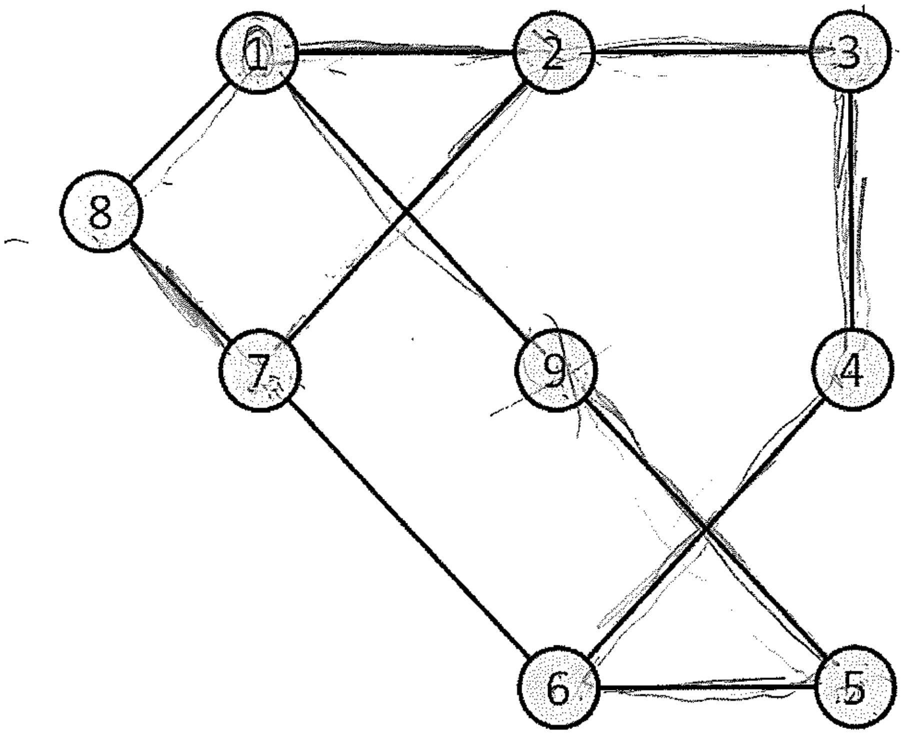
Dark = Red
Light = Black

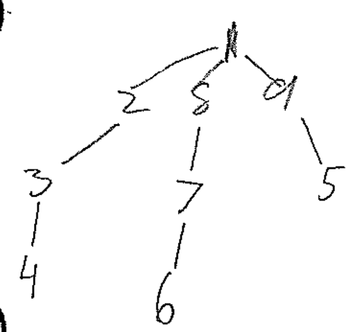


a) Yes
b) No

1. Root Black? Yes
2. If red, then both children black? Yes
3. Equal amount of black nodes to the root? Yes
4 Leafs are all black

9 (10 pts). In a BFS of the graph the nodes are dequeued in a particular sequence for processing. Which of the following node sequences is from a breadth-first search of the graph shown, beginning at node 1:





a) ~~1, 2, 8, 9, 3, 5, 7, 4, 6~~
b) ~~1, 2, 8, 9, 5, 7, 3, 4, 6~~
c) ~~1, 2, 8, 9, 7, 5, 6, 3, 4~~
d) 1, 2, 8, 9, 7, 3, 5, 6, 4
e) ~~All sequences can come from a BFS~~

D is the only answere that follows BFS by using an arbitrarily made algorithm, but keeping it consistant the entire time.

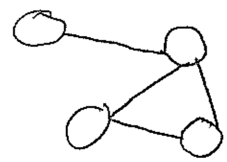
10 (10 pts). In a DFS the nodes are encountered for the first time in a particular sequence. Using the same graph from previous question, which of the following node sequences is not from a depth-first search of the graph shown:

a) 1, 2, 7, 8, 3, 4, 6, 5, 9 ✓
b) 1, 9, 5, 6, 4, 3, 2, 7, 8 ✓
c) 1, 2, 3, 8, 7, 4, 5, 6, 9
d) 1, 8, 7, 2, 3, 4, 6, 5, 9 ✓
e) All sequences can come from a DFS

You need to continue to 4 after visiting 3.

11 (10 pts). Let G be a graph with n vertices. If G has n or more edges, then G must have a cycle.

n verti



a) Always true
b) Sometimes true, other times false
c) Never true

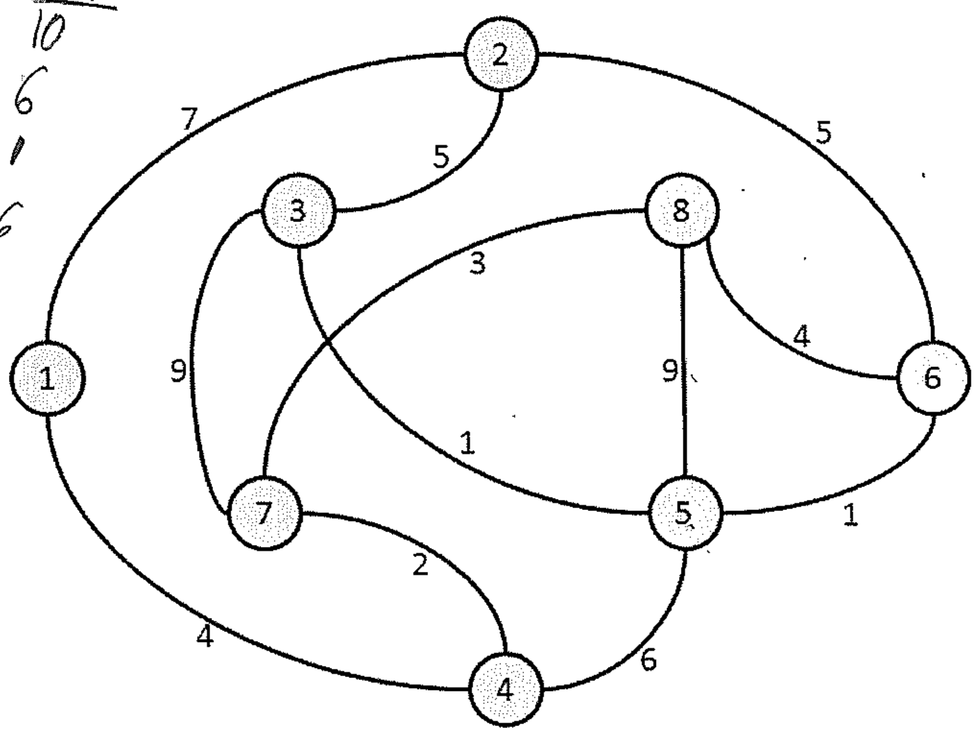
12 (10 pts). Assume a *directed graph* G in which weight of all edges are equal. We need to compute the shortest path from node S to every other node in G. Which of the followings represent time complexity of this computation (n and e representing number of vertices of and number of edges of the graph, respectively):

By using DFS or BFS, we can find that it is O(n+e).

a) $O(n^2)$
b) $O(n * e)$
c) $O(n\ log(e))$
d) $O(n+e)$
e) None of the above

13 (10 pts). Given the graph below, list the vertices along the shortest path (and the total edge sum) to each node starting from node 5.

Nodes	total
1. 5, 4, 1	10
2. 5, 3, 2	6
3. 5, 3	1
4. 5, 4	6
5. 5, 5	0
6. 5, 6	1
7. 5, 4, 7	8
8. 5, 6, 8	5



Submit Instructions:
The homework must be turned in by the due date and time using the turnin command. Follow the next steps:

1. Please make sure your submission is legible! (No cellphone pictures please! All ITAP labs have scanners for you to use)
2. Login to data.cs.purdue.edu (you can use the labs or a ssh remote connection).
3. Make a directory named with your username and copy your solution (in pdf format) there. **(Important:** Such pdf file should be the only one contained within the folder).
4. Go to the upper level directory and execute the following command:
   turnin -c cs251 -p hw3 your_username
   (Important: previous submissions are overwritten with the new ones. Your last submission will be the official and therefore graded).
5. Verify what you have turned in by typing **turnin -v -c cs251 -p hw3**
   (Important: Do not forget the -v flag, otherwise your submission would be replaced with an empty one). If you submit the wrong file you will not receive credit.