Names: March 13, 2019

Computer Science 241

In-Class Review Exercises

1. Circle T or F to indicate whether the statement is true or false.

   (a) T / F The partition step of QuickSort is the "divide" phase of divide-and-conquer, whereas the merge step of MergeSort is the "conquer" phase.

   (b) T / F Finding an element in a binary tree is worst-case O(n).

   (c) T / F Implementing the Set ADT with a linked list would make insertion more efficient than using an array.

   (d) T / F A hash table with a large load factor is more time-efficient but less space-efficient than one with a small load factor.

2. (1 pt) Which of the following **could** be the **result** of a call of the `partition` method in QuickSort?

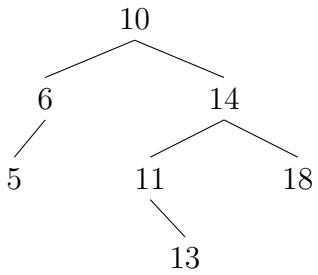
   (a) `[ 2, 5, 2, 4, 1 ]`

   (b) `[ 6, 2, 7, 8, 9 ]`

   (c) `[ 6, 7, 2, 3, 4 ]`

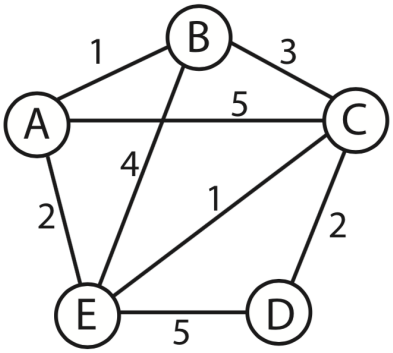
   (d) `[ 7, 9, 3, 4, 5 ]`

3. Consider the following Binary Search Tree:



   (a) Write the sequence of necessary rotations to rebalance the tree, using "direction(value)" to denote a rotation on a node with that value. For example, left(10) indicates a left rotation on the node with value 10.

   (b) Insert 19 into the original tree as drawn above using BST (not AVL) insert.



S F

n	n.d
A	
B	
C	
D	
E	

4. (3 points) Consider the graph above. In the following, break all ties alphabetically (A before B, and so on); recall that BFS and DFS do not look at edge weights.

   (a) (1 point) List nodes in the order visited by `BFS(A)`:

   (b) (1 point) List nodes in the order visited by `DFS(A)`

5. (2 points) Run `dijkstra(A)` on the graph, keeping a record of the order in which nodes are added into the Frontier and Settled sets. You need not keep track of backpointers.

   (a) List nodes in the order added to the Frontier set:

   (b) List nodes in the order added to the Settled set: