Algorithms and Data Structures

Final Exam

August 5, 2025

1. Consider the function `my_function(L)` where `L` is assumed to be a Python list:

```python
def my_function(L):
    for k in range(1, len(L)):
        current = L[k]
        j = k
        while j>0 and L[j-1]>current:
            L[j] = L[j-1]
            j -= 1
        L[j] = current
```

   (a) What is this function doing?

   (b) What is its worst case asymptotic running time as a function of `n = len(L)`?

   (c) What is its best case asymptotic running time? Describe a list `L` for which the best case running time is achieved.

(5+5+5)

2. Sketch, using Python or Python-like pseudocode, the implementation of a function that checks whether a graph $G = (V, E)$ has a cycle. (10)

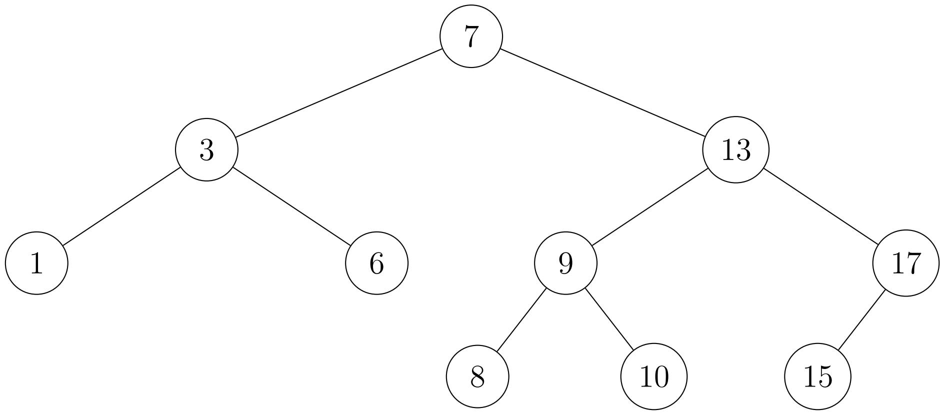
3. Consider the array $[2, 9, 4, 10, 15, 11, 6, 12]$ representing a binary tree in the standard array-representation.

   (a) Draw the corresponding binary tree. Is it a min-heap? If not, restore the min-heap property in a bottom-up pass through the tree.

   (b) Insert the element 1 into the min-heap from part (a) and show all "bubble-up" operations required to restore heap order.

(5+5)

4. Insert the element 16, then the element 18 into the following AVL tree. Indicate all subtree heights before each insertion, and show the changes introduced by the insertions and the subsequent rebalancing step-by-step. (You may consult the summary of tree rotations on the front page.)



(10)

5. We noted in class that we can use the splay tree data structure to sort an array L of length n as follows: Take an initially empty splay tree S and move the elements of L one-by-one into S. Then traverse S in-order, moving the elements back into L. This algorithm is known as "splay-sort".

   (a) Prove that splay-sort takes $O(n \log n)$ time.

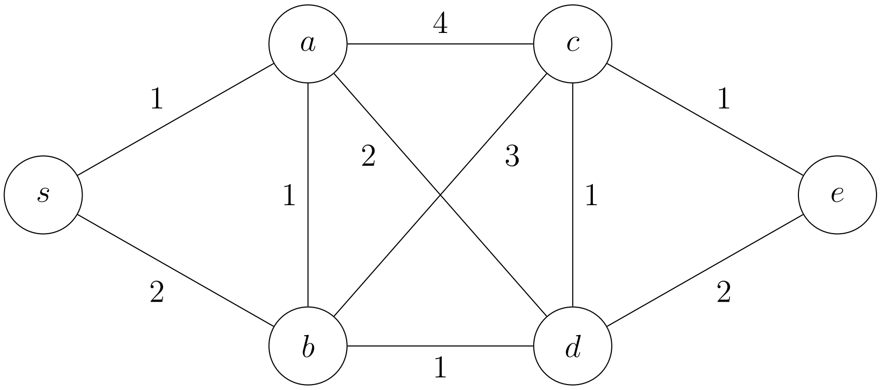
   Note: You can use, without proof, the result from class that m splay tree operations starting from an initially empty splay tree take $O(m \log n)$ time, where n is the maximal size of the tree.

   (b) Prove that splay-sort takes $O(n)$ time when L is already sorted.

(5+5)

6. Let $G = (V, E)$ be a connected weighted graph with non-negative weights. A *shortest-path tree* of G is a tree that contains all the vertices of G such that the path from its root s to every vertex $v \in V$ is a shortest path in G.

   (a) Draw a shortest-path tree for the graph



   (b) Show that a shortest-path tree always exists under the conditions stated.

   (c) Give a small example that shows that a shortest-path tree is not necessarily a minimum spanning tree.

(5+5+5)