Algorithms and Data Structures

Makeup Exam

October 9, 2024

1. (a) Order the following functions by their asymptotic growth rate:

$$n, n + n^2 + n^3, 2^{(n^2)}, (2^n)^2, 2^{2\log n}, 2^{4\log n}$$

Note: $\log n$ *denotes the base-*2 *logarithm!*

(b) Al and Bob are arguing about their algorithms. Al claims his $O(n \log n)$-time method is always faster than Bob's $O(n^2)$-time method. To settle the issue, they perform a set of experiments. To Al's dismay, they find that if $n < 100$, the $O(n^2)$-time algorithm runs faster, and only when $n \geq 100$ is the $O(n \log n)$-time one better. Explain how this is possible.

(5+5)

2. Consider the following algorithm, which takes as input a list `L` containing integer entries:

```python
def f(L):
    n = len(L)
    for i in range(n):
        for j in range (i):
            for k in range(n):
                if abs(L[i]-L[j]) == L[k]:
                    return True
    return False
```

(a) Describe what this algorithm does and state its asymptotic running time in the best case and in the worst case.

(b) Describe an alternative algorithm for the same problem that uses hashing and completes in $O(n^2)$ time. You may assume that inserting a key into the hash table or searching for a key always completes in $O(1)$ time

(5+5)

3. Are the following statements true or false? Explain your answer in 1–2 sentences.

(a) There exists a sort algorithm that can sort a list of length n in $O(n)$ best case.

(b) There exists a sort algorithm that can sort a list of length n in $O(n)$ worst case.

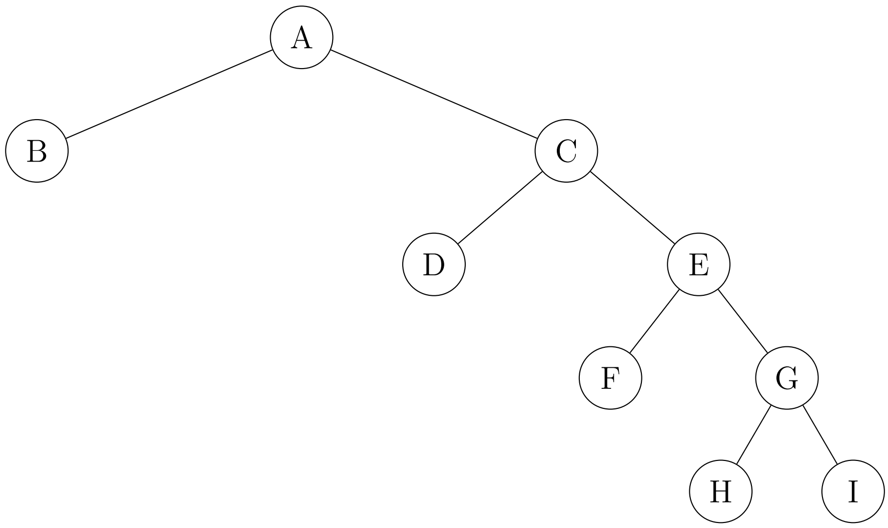
(c) Searching for an element in a splay tree with n elements always completes in $O(\log n)$ time.

(d) Searching for the same element in a splay tree with n elements n times in direct succession takes $O(n)$ time.

(e) When implementing Dijkstra's algorithm to find the shortest path between vertices in a connected graph, we should always use a heap-based adaptable priority queue to get optimal performance.

(2+2+2+2+2)

4. Give the pre-order, in-order, and post-order traversals of the following binary tree:



(5)

5. Insert the keys 7, 3, 6, 9, 4, 5 into an initially empty max-heap. Show the max-heap at each step of insertion. (5)

6. Insert the keys 3, 4, 5, 6, 9, 7 into an initially empty AVL tree tree. Show the AVL tree at each step of insertion.

   (You may consult the attached AVL tree "cheat sheet".) (5)

7. Give a precise and complete description of Kruskal's algorithm for computing the minimum spanning tree of a graph. (5)