Algorithms and Data Structures (IBC027)

January 13, 2021

Whenever an algorithm is required (assignments 2-6), it can be described in pseudocode or plain English. **The explanations of your algorithms may be informal but should always be clear and convincing**. *The grade equals the sum of the scores for all 6 problems below divided by 10 (plus possibly a bonus score for the homework assignments). Good luck!*

1 Quiz (27 points)

Question 1 Is $2^{n+1} \in \mathcal{O}(2^n)$? Is $2^{2n} \in \mathcal{O}(2^n)$?

Question 2 In a binary max heap containing n numbers, the smallest element can be found in time

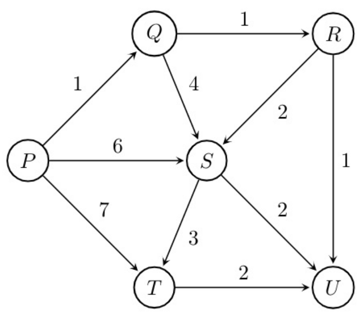
(A) $\mathcal{O}(n)$

(B) $\mathcal{O}(\log n)$

(C) $\mathcal{O}(\log \log n)$

(D) $\mathcal{O}(1)$

Question 3 Suppose we run Dijkstra's shortest-path algorithm on the following edge-weighted directed graph with vertex P as the source.



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized?

(A) P, Q, R, S, T, U

(B) P, Q, R, U, S, T

(C) P, Q, R, U, T, S

(D) P, Q, T, R, U, S

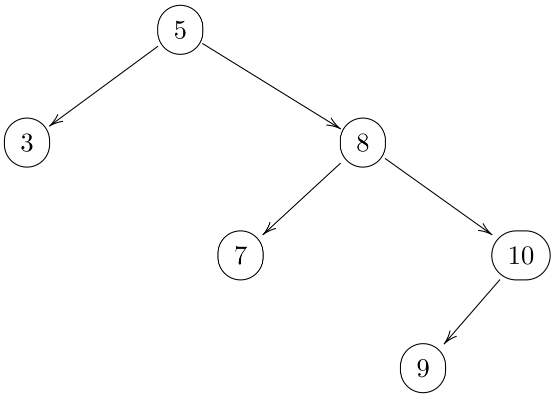
Question 4 Discuss three differences between Dijkstra's algorithm and the Floyd-Warshall algorithm.

Question 5 Consider the following sequence of operations on a binary search tree which is initially empty:

insert(3); insert (1); insert (2); insert(4) insert(5); insert(6); delete(1)

What is the height of the resulting tree?

Question 6 Consider the following binary tree.



Which of the following statements are true? (Multiple answers possible!)

(A) This is a binary search tree;

(B) This can be colored as a red-black tree;

(C) This is an AVL tree;

(D) This is a complete binary tree;

Question 7 Suppose we have a red-black tree, and insert an element into it using the standard procedure for binary search trees. We color the new node red. Which red-black tree properties are potentially violated after this operation?

Question 8 The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \mod 10$ and linear probing. What is the resulting hash table?

(A)

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(C)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(D)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

Question 9 Discuss three advantages/disadvantages of hashing using chaining when compared with hashing using open addressing.

2 Binary search trees (10 points)

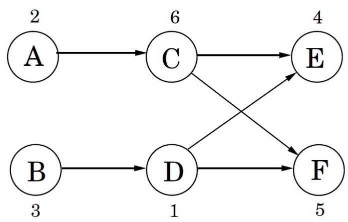
Describe an efficient algorithm for constructing a binary search tree of height $\mathcal{O}(\lg n)$ from a sorted array of n integers. Include a complexity argument.

3 Graph algorithms (16 points)

You are given a directed graph $G = (V, E)$ (in adjacency list format) in which each node $u \in V$ has an associated *price* p_u which is a positive integer. Define an array cost as follows, for each $u \in V$,

$$\mathsf{cost}[u] = \text{price of the cheapest node reachable from } u \text{ (including } u \text{ itself).}$$

For instance, in the graph below (with prices shown for each node), the cost values of the nodes A, B, C, D, E, F are 2, 1, 4, 1, 4, 5, respectively.



Your goal is to design an algorithm that fills the *entire* cost array (i.e., for all vertices).

1. Give a linear-time algorithm that works for directed *acyclic* graphs.
2. Extend this to a linear-time algorithm that works for all directed graphs.

Include a clear explanation and discuss correctness and runtime complexity for both algorithms.

4 Dynamic programming (16 points)

Radboud University plans the construction of a new building. Inspired by the reputation of the Erasmus building as an architectural highlight, they want to build another tower, which has to be as big as possible. But of course space on campus is limited. It is your task to help them find a suitable location, which needs to be a square. Based on a map of the campus, you have to decide the largest square area which is available for this grand new tower.

The input is an $m \times n$ matrix A consisting of zeros and ones, where $A_{i,j} = 1$ means that the location at row i and column j is available, and $A_{i,j} = 0$ means it is not. The problem is to determine the size of the largest square of ones present in this matrix (by size we mean the length of the sides). For instance, if the input A is

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

then the largest such square has size 3. Describe an algorithm for this problem using dynamic programming. Include a clear explanation and a complexity analysis. Hint: compute a matrix S such that $S_{i,j}$ is the size of the largest square of ones in A with (row i, column j) as its bottom right corner.

5 Divide and conquer (16 points)

Let $A = [a_1, a_2, \ldots, a_n]$ be an integer array of length n. An *inversion* is a pair of indices i, j with $i < j$ such that $a_i > a_j$. For instance, in the array

$$[5, 4, 6, 2]$$

there are 4 inversions: $(1, 2), (1, 4), (2, 4)$ and $(3, 4)$.

Describe an efficient algorithm based on divide and conquer, which takes an integer array A as input, and returns the number of inversions in A. Explain your algorithm, and its correctness and runtime complexity.

6 Greedy algorithms (15 points)

Consider Algorithm 1 below, which takes as input an undirected, connected graph $G = (V, E)$ (in adjacency list format) with edge costs c_e:

Algorithm 1: Computation of a minimum spanning tree.

Function `MST(`G`)`
  sort the edges of G according to their weights ;
  for *each edge e from G, in decreasing order of c_e* **do**
    if *e is part of a cycle of G* **then**
      $G \leftarrow G - e$ (that is, remove e from G) ;
  return G ;

---

1. Explain/prove why this algorithm returns a minimum spanning tree.
2. On each iteration, the algorithm must check whether there is a cycle containing a specific edge e. Give a linear-time algorithm for this task and justify its correctness.
3. What is the overall time complexity of this algorithm, in terms of $|E|$?