CONCORDIA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

COMP 6651: Algorithm Design Techniques **Winter 2022**

Quiz # 8

First Name	**Last Name**	**ID#**

Assumptions for union-find algorithm / disjoint set data structure: N **items,** M **operations**

1. Path compression algorithm performs in which of the following operations?

   A. Create operation

   B. Insert operation

   C. Find operation

   D. Delete operation

   Your choice: A ☐ B ☐ C ☒ D ☐

   1 point. C. Path compression algorithm is performed during find operation and is independent of the strategy used to perform unions.

2. What is the worst case efficiency for a path compression algorithm?

   A. $O(N)$

   B. $O(\log N)$

   C. $O(N \log N)$

   D. $O(M \log N)$

   Your choice: A ☐ B ☐ C ☐ D ☒

   1 point. D.

3. What is the depth of any tree if the union operation is performed by height (no path compression)?

   A. $O(N)$

   B. $O(\log N)$

   C. $O(N \log N)$

   D. $O(M \log N)$

   Your choice: A ☐ B ☒ C ☐ D ☐

   1 point. B. If the unions are performed by height, the depth of any tree is $O(\log N)$. The idea is to make the shorter tree a child of the root of the taller in the union operation. If tree T_1 is strictly shorter than T_2, and if T_1 is made a child of the root of T_2, then the overall height of the resulting tree does not change. If the two trees are the same height, then the height increases by 1, but this is the only way the height can increase. Therefore worst case is binary tree, and therefore a depth of $O(\log N)$.

4. Consider the following program

    for i from 1 to N
      MakeSet(i)
    For i from 1 to $N-1$
      Union($i, i+1$)

Assume that the disjoint set data structure is implemented as disjoint trees with union by rank heuristic

What is the number of trees in the forest and the maximum height of a tree in this forest after executing this code? (Recall that the height of a tree is the number of edges on a longest path from the root to a leaf. In particular, the height of a tree consisting of just one node is equal to 0.)

A. One tree of height $\log_2 N$

B. Two trees, both of height 1

C. $N/2$ trees, the maximum height is 2

D. $\log_2 N$ trees, the maximum height is 1

E. N trees, the maximum height is 1

F. One tree of height 1

Your choice: A ☐ B ☐ C ☐ D ☐ E ☐ F ☒

2 points. F.

5. The off-line minimum problem asks us to maintain a dynamic set T of elements from the domain $\{1, 2, \ldots, n\}$ under the operations INSERT and extract-min. We are given a sequence S of n insert and m extract-min calls, where each key in $\{1, 2, \ldots, n\}$ is inserted exactly once. We wish to determine which key is returned by each extract-min call. Specifically, we wish to fill in an array extracted$[1..m]$, where for $i = 1, 2, \ldots, m$, extracted$[i]$ is the key returned by the ith extract-min call. The problem is "off-line" in the sense that we are allowed to process the entire sequence S before determining any of the returned keys

In the following instance of the off-line minimum problem, each insert is represented by a number and each extract-min is represented by the letter E:
4, 8, E, 3, E, 9, 2, 6, E, E, E, 1, 7, E, 5.
Fill in the correct values in the extracted array: indicate below the output of each insert extracted operation

3 points for a complete exact filling of row extracted. No point for the filling of row insert as the definition of insert was not recalled (it is in question (b) in the coursepack).

	1	2	3	4	5	6	7
extracted	4	3	2	6	8	1	
insert	$I_1 = \{4, 8\}$	$I_2 = \{3\}$	$I_3 = \{9, 2, 6\}$	$I_4 = \emptyset$	$I_5 = \emptyset$	$I_6 = \{1, 7\}$	$I_7 = \{5\}$

6. Recall the definition of a Polynomial Time Approximation Scheme (PTAS) for a Minimization Problem

2 points. A PTAS is an algorithm which takes an instance of an optimization problem and a parameter $\varepsilon > 0$ that is within a factor $1+\varepsilon$ of being optimal for maximization or minimization problems according to the definition given in the slides of the lecture. In addition, algorithm has to be with a complexity that is polynomial in the problem size for every fixed ε.
1 point to express **clearly** the idea of approximation algorithm, **1 point** for the polynomial complexity in the problem size for fixed ε.