THE UNIVERSITY OF SYDNEY

CONFIDENTIAL EXAM PAPER

This paper is not to be removed from the exam venue.

Computer Science

EXAMINATION

Semester 1- Main, 2021

COMP2123 Data Structures and Algorithms

EXAM WRITING TIME: 2 hours
READING TIME: 10 minutes

EXAM CONDITIO
This is a OPEN boo
All submitted worl g someone else's help, in acc onesty and Plagiarism" policies

MATERIALS PERN

MATERIALS TO B

INSTRUCTIONS T
Type your answers ert it into a pdf file.

Submit this pdf file ted. Handwritten responses w

Start by typing you submission. Do **not** type your name.

Submit only your answers to the questions. Do **not** copy the questions.

Do **not** copy any text from the permitted materials. Always write your answers in your own words.

For examiner use only:

Problem	1	2	3	4	Total
Marks					
Out of	10	10	20	20	60

Problem 1.

a) Suppose we have a priority queue PQ, implemented as a min-heap, containing n keys and some integer x. [5 marks]

```
def FOO(x)
    result ← 0
    while PQ.min() < x^2 do
        temp ← PQ.remove_min()
        result ← result + temp^2
    return result
```

Analyze the time complexity of running FOO.

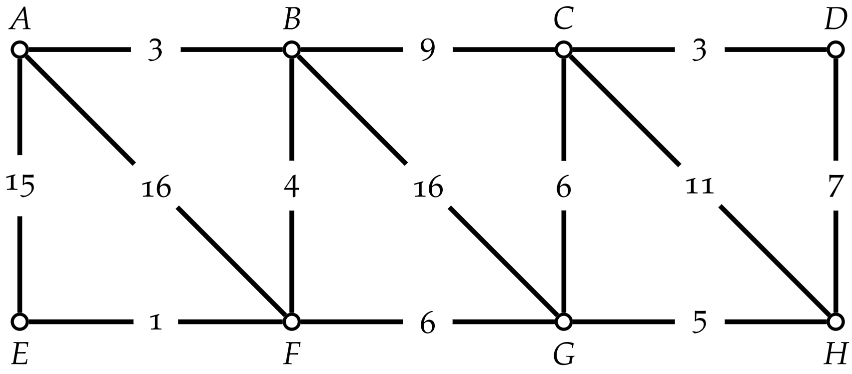
b) We are given a set of items $I = \{i_1, ..., i_n\}$ and sets $S_1, ..., S_m$ containing subsets of these items, i.e., $S_k \subseteq I$ for all $1 \le k \le m$. The sets don't have to contain the same number of items and an item may occur in multiple sets. We need to find the smallest se [illegible] ement from each set S_k ($1 \le k \le$ [illegible] [5 marks]

Example:
$I = \{i_1, ..., i_6\}$
$S_1 = \{i_1, i_2, i_6\}$
$S_2 = \{i_2, i_4, i_5,$ [illegible]
$S_3 = \{i_2, i_4\}$
In the exampl [illegible] set of items such that we have a [illegible] f all three sets S_1, S_2, and S_3. Th [illegible] each of the three sets, but it isn [illegible] ns two elements and $\{i_2\}$ conta [illegible]

Construct a co [illegible] hm doesn't compute the *small* [illegible] hat contain them (for ease of wr [illegible] contains the item i_j) in decreasi [illegible] $_k$ and no item in T is an element of S_k, we add i_j to T.

```
def SMALLESTSET(S_1, ..., S_m, I)
    T ← [ ]
    Sort I by the number of sets that contains each item in decreasing
    order and renumber such that |i_1| ≥ |i_2| ≥ ... ≥ |i_n|
    for j ← 1; j ≤ n; j++ do
        if i_j is part of some S_k and no other item in T is part of S_k then
            T ← T ∪ i_j
    return T
```

Problem 2. Consider the following edge weighted undirected graph:



Your task is to:

a) Compute the shortest path tree T of the graph starting from A. List the edges in T. [7 marks]

b) Indicate the or gorithm starting from A. [3 marks]

(You do **not** have to

Problem 3. Recall that a forest is a graph where every connected component is a tree. We want to design a data structure for a fixed set of n vertices that allows us to add and remove edges, as well as efficiently answer the query: "Are vertex i and vertex j part of the same tree?" You can assume that we identify the vertices by their number and that each vertex has a unique number in the range 0 to $n-1$ (or 1 to n if that's easier for you).

Your data structure should support the following operations:

- INITIALIZE(n): construct the data structure for the n vertices without any edges. This method is called exactly once and only as the first operation in any execution.
- INSERT-EDGE(i, j): insert an undirected edge between vertex i and vertex j, if adding this edge doesn't create cycles (otherwise, don't add the edge)
- REMOVE-EDGE(i, j): remove the edge between vertex i and vertex j, if it exists
- IN-SAME-TREE(… of the same tree, and False othe…

Example:
INITIALIZE(10) - crea…
IN-SAME-TREE(9,2) - …
INSERT-EDGE(2,6) - a…
INSERT-EDGE(9,6) - a…
INSERT-EDGE(9,2) - c…
IN-SAME-TREE(9,2) - …
REMOVE-EDGE(6,2) - …
IN-SAME-TREE(9,2) - …

Your task is to … space. The INITIALIZE, INSERT-EDG… $O(n^2)$ time. The IN-SAME-TREE opera…

a) Describe your … [8 marks]

b) Prove the correctness of your data structure. [7 marks]

c) Analyze the time and space complexity of your data structure. [5 marks]

Problem 4. Silbo Gomero is a whistling language used in the border region of France and Spain by shepherds to communicate with each other. We want to determine the number of pairs of shepherds that can communicate using this language. More specifically, we are given the locations of the shepherds in an array A, where every location is represented by a distinct positive integer. For simplicity you can assume that every shepherd whistles equally loudly and thus can cover the same distance d. Shepherds i and j can communicate with each other if $|A[i] - A[j]| \leq d$, i.e., if the absolute difference between their locations is at most the distance they can cover by whistling. Recall that $|x|$ equals x when $x \geq 0$ and $|x|$ equals $-x$ if $x < 0$. You need to determine how many pairs of shepherds can communicate with each other.

Example:
When $A = [4, 2, 12, 7]$ and $d = 3$, you should return 2, since $|4 - 2| = 2 \leq d$ and $|4 - 7| = 3 \leq d$ and no other pair is at distance at most d from each other.

Your task is to give a divide and conquer algorithm for this problem that runs in $O(n \log n)$ time. Re

a) Describe your [8 marks]

b) Prove the corr [7 marks]

c) Analyze the ti [5 marks]

(Disclaimer: Silb und information given in the questio