Name of Candidate: ____________________________

Student ID: _____________________________________

Signature: ______________________________________

UNSW Sydney
Term 1 2020 Final Examination
COMP9024 – Data Structures and Algorithms

- **Time Allowed:** 2 Hours
- **Reading Time:** 10 minutes
- **Total Number of Pages:** 5 (including title page)
- **Total Number of Questions:** 9
- **Total Marks Available:** 100 (marks available for each question are shown in the examination paper)
- Answer all questions.
- Write your answers in the exam booklets provided.
- All answers must be written in ink except where they are expressly required. Pencils may be used only for drawing, sketching or graphical work.
- **The following materials will be provided**: 2x 8-page exam booklets.
- This paper may **not** be retained by the candidate.

PART I: Basic Data Structures and Algorithms (40 marks)

Question 1 (**8 marks**) Consider the following algorithm which takes an array A of n integers as input and uses an initially-empty queue Q as a local variable:

```
Algorithm Unknown(A)
   Input: A one-dimensional integer array A of size n
   Output: To be determined
 {
   create an empty queue Q;
   enqueue(Q, A[0]);  // enqueue(Q, x) adds x to the end of Q
   for ( i = 1; i<= n – 1; i++ )
     {
       if ( A[i] >=A[i-1] )
          enqueue(Q, A[i]);
       else
         {
            t=0;
            while ( Q is not empty )
               t = t + dequeue(Q);
              // dequeue(Q) removes the first element of Q and returns it
            output t;     // print t on the standard output
            enqueue(Q, A[i]);
         }
     }
    if ( Q is not empty)
     {
        t=0;
        while ( Q is not empty )
            t = t + dequeue(Q);
        output t;  // print t on the standard output
     }
 }
```

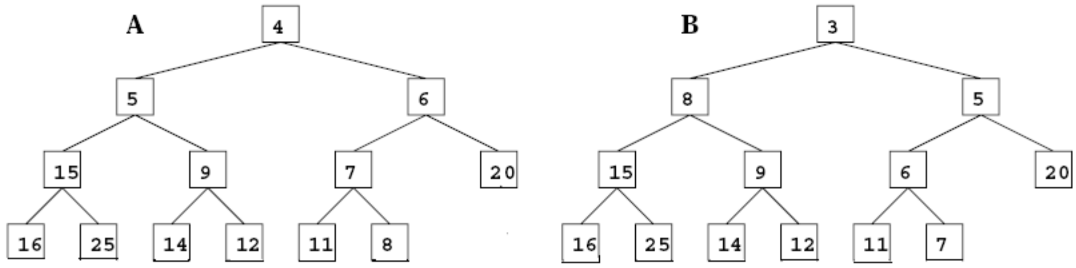
Please answer each of the following questions concerning this algorithm:

(a) What are printed on the standard output when this algorithm terminates for the array A = {1, 9, 12, 12, 14, 5, 6, 8, 13, 3, 12, 5, 5}? (**2 marks**)

(b) Describe what this algorithm prints on the standard output. (**1 marks**)

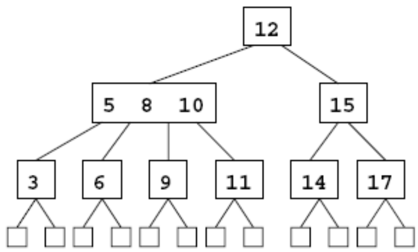
(c) Characterize, using the big-O notation, the running time of the above algorithm in terms of n, the number of integers in A. (**5 marks**)

Question 2 (**8 marks**) What minimal sequence of insert and/or removeMin operations on heap A will transform it into heap B? Draw the heap after each operation.

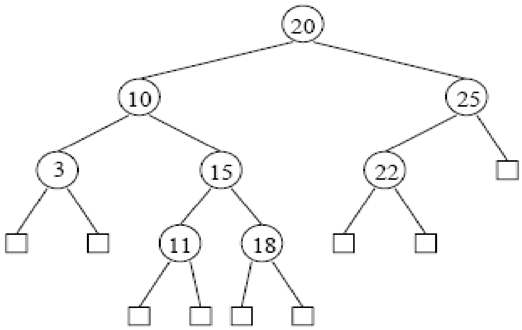


Question 3 (**8 marks**) Let T be a (2, 4) tree shown below, which stores items with integer keys. Draw all the steps (splitting, fusing, and transfer) and the resulting trees obtained by performing the following operations on T :

1. Insert three items with keys 1, 2, 4 in this order into T. (**4 marks**)
2. Remove the item with key 14 from the tree resulting from the insertions of part 1. (**4 marks**)



Question 4 (**8 marks**) In the search for an element x in a binary search tree, the list of nodes that x is compared to is called the key *sequence* for the search. For example, the key sequence for the search for 18 in the following tree is 20, 10, 15, 18.



Suppose that we have the numbers 1 to 1000 in a binary search tree and we want to search for the number 750. Explain why the following could not be the key sequence for the search: 925, 502, 711, 805, 500, 750.

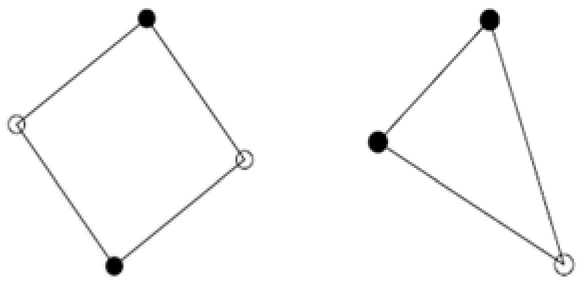
Question 5 (**8 marks**) Draw the frequency array and Huffman tree for the following strings "data structures and algorithms".

PART II: Design and Analysis of Algorithms (60 marks)

Question 6 (**15 marks**) Given a sequence of n numbers, the k ($k \leq n$) smallest numbers of the sequence are the first k numbers of the sequence sorted in non-decreasing order. For example, if the sequence is 35, 12, 25, 4, 50, 28, 23, 62, 55, 12, 4, the 6 smallest numbers are 4, 4, 12, 12, 23, 25. Describe an O(n+k*log n)-time algorithm for finding the k smallest numbers of a sequence of n numbers. Also show why your algorithm takes O(n+k*log n)-time. Your algorithm can call any algorithms in the lecture notes without giving their pseudo code.

Question 7 (**15 marks**) Let T be a text of length n, and let P be a pattern of length m. Describe an O(n+m)-time algorithm for finding the longest prefix of P that is a substring of T, and show why your algorithm takes O(n+m) time. For example, if T = dataanddatastorage and P = datastructures, then the longest prefix of P that is a substring of T is datast. Your algorithm can call the algorithm for computing the failure function without giving its pseudo code.

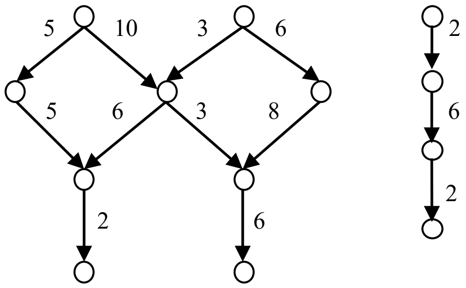
Question 8 (**15 marks**) Describe an O(n+m) time algorithm for testing if an undirected graph is a bipartite graph, where n is the number of vertices and m is the number of edges of the graph. Also explain why your algorithm takes O(m+n) time. You need to give complete pseudo code for your algorithm.



(a) 2-colorable (b) not 2-colorable

Question 9 (**15 marks**) A DAG (Directed Acyclic Graph) is a directed graph without any cycles. An edge-weighted graph is a graph where each edge has a weight. The path length of a path of an edge- weighted graph is the sum of all the constituent edge weights of the path.
Design an O(n+m) time algorithm for computing the longest path length of a weighted DAG, where n is the number of vertices and m is the number of edges of the DAG. Also explain why your algorithm has the time complexity it does. The longest path length of a weighted DAG is the largest path length of all the paths starting at a source node and ending at a sink node. A source node is a node with 0 indegree, and a

sink node is a node with 0 outdegree. For example, the longest path length of the following edge-weighted DAG is 20 (6+8+6).



Your algorithm can call any algorithms given in the lecture notes without giving their pseudo code.

END OF EXAMINATION PAPER