Module: INT102 **Assignment 1**

1. Assessment

The tasks contribute 10% to the overall assessment of INT102

2. Submission

Please complete the assessment tasks using Microsoft Word and submit it in PDF via Learning.

3. Deadline

19-April- 2021, Monday 17:30.

Question 1 [10 marks]

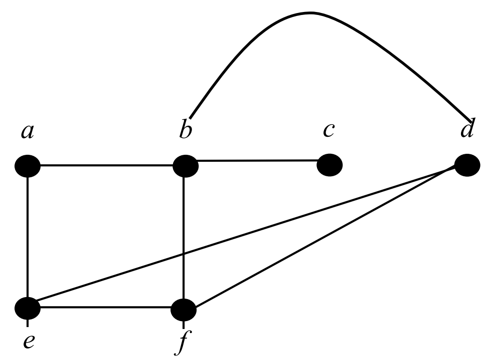
Consider the function: $f(n) = 3n + n^2 + 2n \log n + 2\sqrt{n}$

1. State the order of magnitude (in Big-O notation) of the function. **(5 marks)**

2. Prove that the function *f(n)* is of the order of magnitude as you stated above. **(5 marks)**

Question 2 [10 marks]

Consider the following graph G.



1. Starting at the vertex ***a*** and resolving ties by the vertex alphabetical order traverse the graph by breadth-first-search (BFS) and construct the corresponding BFS tree. **(5 marks)**

2. Starting at the vertex ***a*** and resolving ties by the vertex alphabetical order traverse the graph by depth-first-search (DFS) and construct the corresponding DFS tree. **(5 marks)**

Question 3 [10 marks]

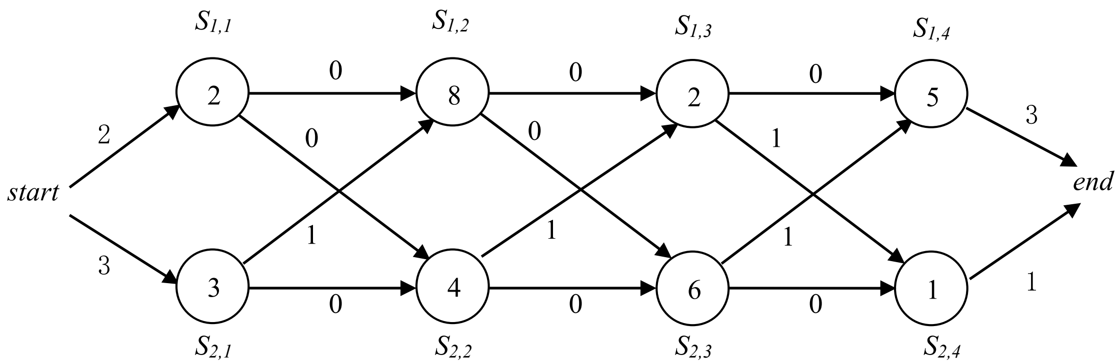
Given the Bubble sort algorithm as below:

```
ALGORITHM BubbleSort(A[0..n − 1])
//Sorts a given array by bubble sort
//Input: An array A[0..n − 1] of orderable elements
//Output: Array A[0..n − 1] sorted in ascending order
for i=0 to n − 2 do
    for j = n-1 downto i+1 do
        if A[j ]<A[j-1 ] swap A[j ] and A[j - 1]
```

1. What is the number of swapping operations needed to sort the numbers A[0..5]=[3，4，5，3，4，5] in ascending order using the Bubble sort algorithm? **(5 marks)**
2. What is the number of key comparisons needed to sort the numbers A[0..5]= [3，4，5，3，4，5] in ascending order using the Bubble sort algorithm? **(5 marks)**

Question 4 [20 marks]

Extend the assembly line scheduling problem by adding the transfer time from the start to S_{i1} (i = 1,2) as $t_{0,i}$ and the transfer time from S_{in} (i = 1,2) to the end as $t_{i,n}$.. An extended assembly line with 4 stations is show in the figure below.



1. Write down the recursive formula to compute $f_i[j]$ (i = 1, 2; j= 1, …, n) and f^*. **(5 marks)**

2. Refer to the above extended assembly line instance, fill in the following table. **(5 marks)**

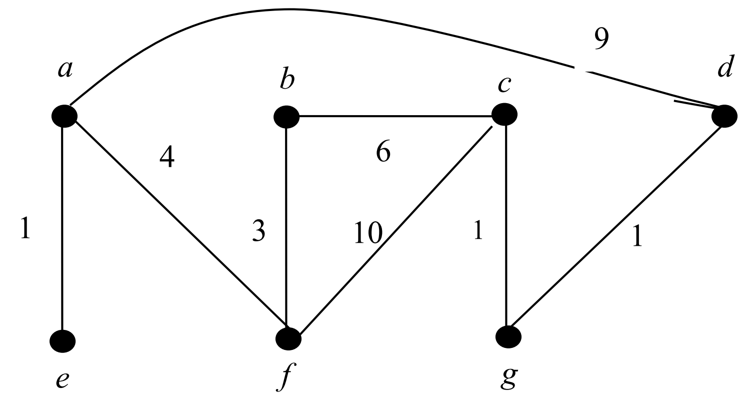
j	$f_1[j]$	$f_2[j]$
1		
2		
3		
4		

3. Using dynamic programming, write a pseudo code to compute f^*. What is the time complexity of your algorithm? **(5 marks)**

4. Based on the line information on the table, describe how to find the fastest way (which stations should be chosen?) **(5 marks)**

Question 5 [20 marks]

Consider the following graph G. The label of an edge is the cost of the edge.



1. Using *Prim's* algorithm, draw a *minimum spanning tree* (MST) of the graph Also write down the change of the priority queue step by step and the order in which the vertices are selected. **(7 marks)**

2. Using *Kruskal's* algorithm, draw a *minimum spanning tree* (MST) of the graph G. Write down the order in which the edges are selected **(5 marks)**

3. Referring to the same graph above, find the shortest paths from the vertex ***a*** to *all* other vertices in the graph G using *Dijkstra's* algorithm. Show the changes of the priority queue step by step and give the order in which edges are selected. **(8 marks)**

Question 6 [10 marks]

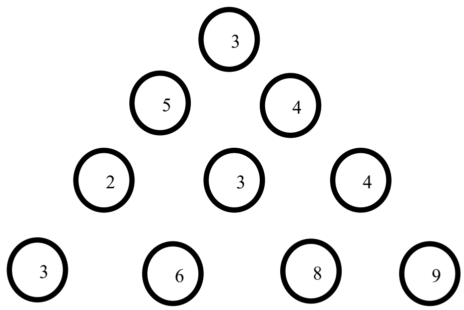
1. Write pseudocode for a divide-and-conquer algorithm for the exponentiation problem of computing a^n where n is a positive integer. **(4 marks)**

2. Set up and solve a recurrence relation for the number of multiplications made by this algorithm. **(4 marks)**

3. How does this algorithm compare with the brute-force algorithm for this problem? **(2 marks)**

Question 7 [20 marks]

Minimum-sum descent: Some positive integers are arranged in an equilateral triangle with n numbers in its base like the one shown in the figure below for $n = 4$. The problem is to find the smallest sum in a descent from the triangle apex to its base through a sequence of adjacent numbers (shown in the figure by the circles). You are asked to solve the problem using a dynamic programming algorithm.



1. Describe and define the sub-problem(s). **(5 marks)**
2. Write down a formula that relates the solution of the problem with solution(s) of sub-problem(s). **(5 marks)**
3. Using dynamic programming, write pseudocode to compute the smallest sum of minimum-sum descent. What is the time complexity of your algorithm? **(5 marks)**
4. What is the smallest sum of the above minimum-sum descent problem? **(5 marks)**