80

(*University of Choice*)

MASINDE MULIRO UNIVERSITY OF SCIENCE AND TECHNOLOGY

(MMUST)

MAIN CAMPUS

UNIVERSITY EXAMINATIONS

END OF SEMESTER EXAMINATION

2022/2023 ACADEMIC YEAR

THIRD YEAR FIRST SEMESTER EXAMINATION

FOR THE DEGREE IN

(COMPUTER SCIENCE)

COURSE CODE: BCS 312

COURSE TITLE: DESIGN AND ANALYSIS OF ALGORITHMS

DATE: 14/12/2022 **TIME: 8:00-10:00AM**

INSTRUCTIONS TO CANDIDATES:

ANSWER QUESTIONS ONE AND ANY OTHER TWO.

Paper Consists of 6 Printed Pages. Please Turn Over

QUESTION ONE (COMPULSORY) [30 MARKS]

a. Explain the characteristics of the following algorithms and how they are applied.

i. Branch and Bound **[2 marks]**

ii. Dynamic Programming **[2 marks]**

iii. Backtracking **[2 marks]**

b. Explain THREE reasons why the study of algorithm is of interest to computing students? **[3 marks]**

c. Why do we require the understanding of complexity theory when handling computational problems? **[3 marks]**

d. Explain the control abstraction of divide and conquer technique **[2 marks]**

e. You are given the following array elements in Table 1 below.

Table 1: Array Elements in Computer Memory

101	103	106	107	119	126	128	180	210	290	300	320	410

Explain how you will search for a value x=200 using binary search and estimate its amortized case and the worst case scenarios. **[4 marks]**

f. i. Define the meaning of a spanning Tree and a Minimum Spanning Tree (MST). **[3 marks]**

ii. Write an algorithms for finding a MST using Kruskal's algorithm and apply it on the network diagram in Figure 1 below? **[5 marks]**

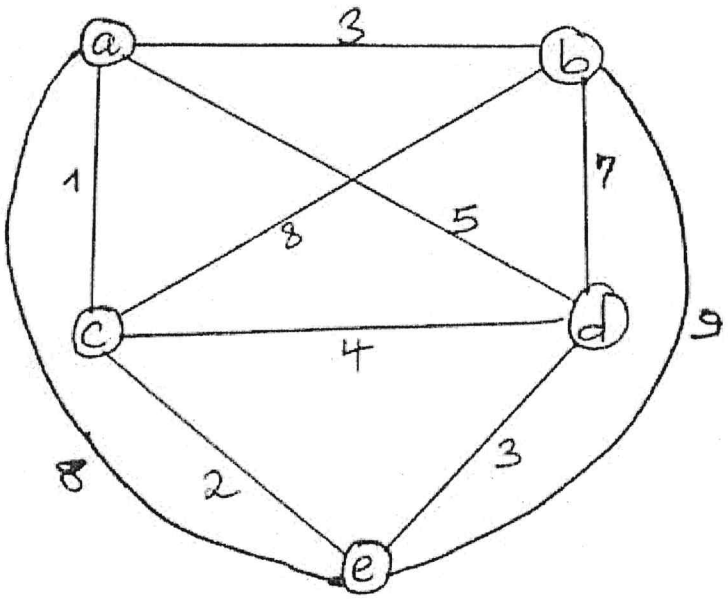

Figure 1: Network Diagram 1.

g. The code segment below is used to compute the matrix product C = A * B. study it and answer the question that follows.

```
for (i = 1; i<=n; i++)
 for (j = 1; j<=n; j++)
   C[i, j] = 0; 1 n*n
   for (k = 1; k<=n; k++)
     C[i, j] = C[i, j] + A[i, k] * B[k, j];
```

Compute the running time of the code in Big-oh notation. **[4 marks]**

QUESTION TWO [20 MARKS]

a. Using rules of sums and rules of products show that f (n) $=4n^3 + 10n^2 + 5n + 1$ is in g (n) is in $O(n^4)$ **[2 marks]**

b. How will you classify a given problem as P and NP type? Discuss the available methods to solve NP problems. **[4 marks]**

c. What is a greedy technique? Explain its important parameters. **[4 marks]**

ii. Consider a knapsack of capacity, W=75 and the list of provided items as in the Table 2 below.

Table 2: Items to be added in the Knapsack

Items	1	2	3	4	5
Profits	180	100	120	120	140
Weights	40	10	20	24	20

Compute the optimized solution or maximum profit in the knapsack using:

i. Greedy strategy **[3 marks]**

ii. Dynamic programming approach **[3 marks]**

iii. Which strategy will you prefer? Justify. **[1 mark]**

d. What is the smallest value of n such that an algorithm who's running times is $\mathbf{100n^2}$ runs faster than an algorithm whose running time is **2n** on the same machine? **[3 marks]**

QUESTION THREE [20 MARKS]

a. Discuss the applicability of design and analysis of an algorithm to a computer science student. **[3 marks]**

b. Explain Euclid's algorithm for computing gcd(m, n), hence compute the gcd and the lcm of (31415 and 14142) **[4 marks]**

c. Discuss the basic NP-complete problems showing their transformation topology.**[3 marks]**

d. Let M be a deterministic Turing Machine (TM) that halt on all inputs. Space complexity of M is a function f: N→N, where f (n) is the maximum number of call of tape and M scans any input of length n. if the space complexity of M is f (n), then M runs in space f (n). Estimate the space complexity of TM using asymptotic notation. **[3 marks]**

e. Discuss Prism's algorithm and use it to estimate the minimum cost of the network diagram in Figure 1. **[3 marks]**

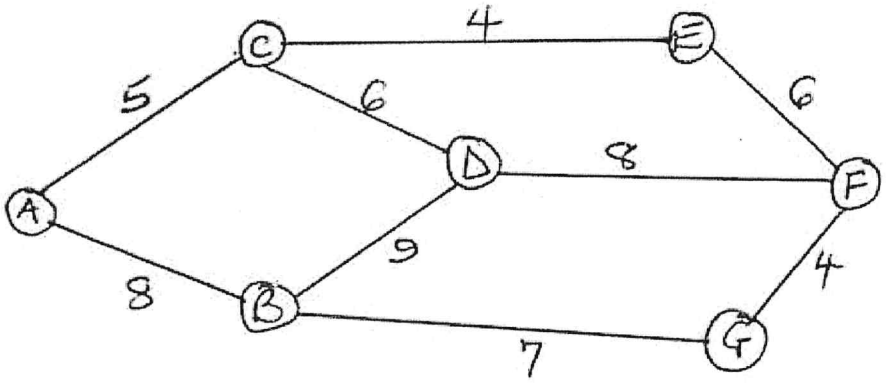

Figure 1: Network Diagram 1.

i. Consider the following recursive algorithm for computing the sum of the first n cubes.

```
S(n)=1^3+2^3+3^3+. . . +n^3
Algorithm S(n)
    If (n=1) return 1
      Else return (S (n-)+ n*n*n))
 end algorithm
```

Set up and solve a recurrence relation for the number of times the basic operation of the algorithm is executed. **[4 marks]**

QUESTION FOUR [20 MARKS]

a. Explain important fundamental problem type of different categories. **[4 marks]**

b. Explain in brief the basic asymptotic efficiency class **[4 marks]**

c. Explain the method of comparing the order of growth of two functions using limits. **[2 marks]**

d. Consider the graph shown in Figure 2 below.

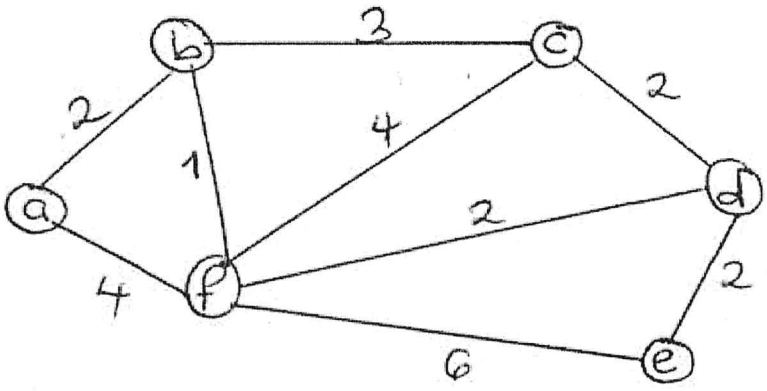

Figure 2: Network Diagram 2.

e. Using appropriate algorithms:

ii. Construct the minimum Spanning Tree (MST). **[3 marks]**

iii. Single source shortest path from node **a** to **e**. **[3 marks]**

f. The aim of designing algorithms is to find an algorithm whose upper bound matches the lower bound of the problem. Write down the procedure used to design an algorithm. **[4 marks]**

QUESTION FIVE [20 MARKS]

a. Discuss various factors that may affect the running time of a computer program. **[3 marks]**

b. Describe how Randomized algorithm the algorithms work. **[3 marks]**

c. Apply Branch and Bound algorithm on the figure 3 below to solve the Traveling Salesman problem. **[3 marks]**

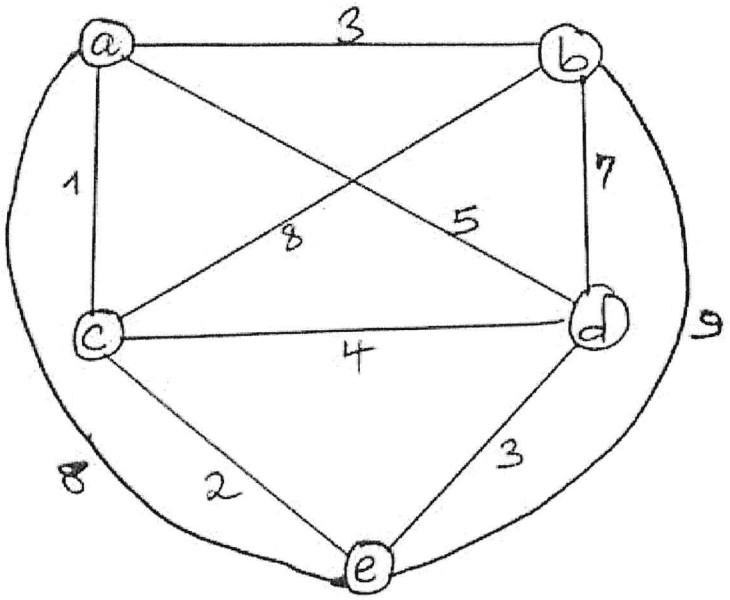

Figure 3: Network Diagram 3.

d. Software packages A and B of complexity O (n log n) and O(n), respectively spend exactly TA (n)= CA n log n and TB (n) = CB n milliseconds to process n data items. During a test, the average time of processing n = 10000 data items with packages A and B is 100 milliseconds and 1500 milliseconds, respectively. Derive exact conditions when one package actually outperforms the other and recommend the best choice if up to $n=10^9$ data items should be processed. **[3 marks]**

e. Let processing time of an algorithm of Big-Oh complexity O(f(n)) be directly proportional to f(n). Let three such algorithms A, B, and C have time complexity $O(n^2)$, $O(n^{1.5})$, and O(n log n), respectively. During a test, each algorithm spends 10 seconds to process 100 data items. Derive the time each algorithm should spend to process 10,000 items. **[3 marks]**

f. Consider the following recursive algorithm for computing the sum of the first n cubes.

```
S(n)=1^3+2^3+3^3+. . . +n^3
Algorithm S(n)
    If (n=1) return 1
      Else return (S (n-)+ n*n*n))
end algorithm
```

Set up and solve a recurrence relation for the number of times the basic operation of the algorithm is executed. **[5 marks]**