National University of Computer and Emerging Sciences

Karachi Campus

Design and Analysis of Algorithms (CS2009)

Final Exam Paper Part-A

Date: Dec 23rd 2024

Total Time (Mins): 30

Total Marks: 10

Total Questions: 2

Course Instructor(s)

Dr. Muhammad Atif Tahir, Dr. Nasir Uddin, Dr. Kamran, Dr. Fahad Sherwani, Dr. Farrukh Salim Shaikh, Ms. Anaum Hamid, Mr. Syed Faisal Ali, Mr. Sandesh Kumar, Mr. Minhal Raza, Mr. Abu Zohran

_____ _____ _____

Roll No Section Student Signature

CLO #2: To analyze complexities of different algorithms using asymptotic notations, complexity classes, and standard complexity function

Question 1: **[0.125 for each column * 4 = 0.5/part, 0.5 *10 = 5 Marks]**

Problems	Algorithm Identified	Worst Case Complexity	Space	Design Technique and Data Structures
Set of critical servers or routers that need to be monitored or secured to ensure all connections are covered.	Vertex Cover	$O(2^n x\ n^2)$ $/O(n+m)$	$O(n)/O(n+m)$	NPC Approximation Algo/Brute Force Graphs with Adjacency List
To graduate, students must take courses in a sequence that respects prerequisite relationships. To determines the order in which courses should be	Topological Sort	O(E+V)	O(V)	Reverse DFS Traversal/Greedy Approach/Graph with Adjacency list
Multiplying adjacency matrices to compute paths of different lengths.	Matrix Chain Multiplication	$O(n^3)$	$O(n^2)$	Dynamic Programming with 2D Arrays
Optimizing the placement of environmental sensors to cover all critical areas with the fewest sensors possible.	Set Cover	$O(2^m x\ m)$ $/O(m.n)$	O(m+n)/O(m)	Brute Force/Greedy Approximation NPC Algorithm with Arrays
Detect words that are like those in the dictionary, enabling efficient spelling correction.	KMP Algorithm	O(n+m)	O(m)	Brute Force with Arrays
To allocate non- conflicting tasks or jobs to resources, ensuring that no two tasks that share common resources are assigned to the same time slot.	Independent Set	$O(2^n)/O(n)$	O(n)	Brute Force/Greedy Approximation Graphs with Adjacency List
To find intersections between roads or pipelines, which is important for optimizing traffic flow or planning infrastructure	Line Intersection Algorithm	$O(n^2)$	O(n)	Geometric Algorithm CCW Approach with Sets/Array's/Vectors

Searching for a target element in a sorted dataset (e.g., a contact in a sorted phonebook).	Binary Search	O(logn)	O(logn)	Divide and Conquer with Arrays
Organizing a shuffled deck of numbers into ascending order.	Merge Sort	O(nlogn)	O(nlogn)	Divide and Conquer with Arrays
Choosing a mix of food items that meet nutritional requirements within a budget or	KnapSack Problem	O(n*w)	O(n*w)	Dynamic Programming with 1D and 2D Arrays

CLO #2: To analyze complexities of different algorithms using asymptotic notations, complexity classes, and standard complexity function

Question 2 (A): [3 Points] For each of the following questions, indicate whether it is T (True) o r F (False) and justify using some examples e.g. assuming a function?

1. $f(n) = O(g(n))$ *if and only if* $g(n) = \Omega(f(n))$

   If f(n) = n and g(n) = n^2 then n is O(n^2) and n^2 is Ω(n)

   True

2. *if* $f(n) \geq 0$ *and* $g(n) > 0$, *then* $f(n)$ *can belong to* $O(g(n)), \Omega(g(n)),$ *or* $\theta(g(n))$

   if f(n) = n and g(n) = n^2 then f(n) ϵ O(g(n)) because $n \leq n^2$ for large n

   True

3. *if* $f(n) = g(n) + h(n)$ *and* $g(n) \in \theta(h(n))$, *then* $f(n) = \theta(g(n))$

   g(n) = n and h(n) = n^2 then g(n) = $\theta(h(n))$ because n grows slower than n^2 and both functions are asymptotically related. F(n) = g(n) + h(n) = n + n^2

   False

Question 2 (B): [2 Points] Compute the time complexity of the following. Show all steps clearly

1.
```
for(int i = 0; i<=(n-2)/3; i++)
        print("Hi");
```
Solution: O(n)

2.
```
Mystery_Algo(a, b):
    result = 1
    while b > 0:
        if b % 2 == 1:  # If b is odd
            result *= a
        a *= a
        b //= 2
    return result
```
Solution: O(log b)