|  |  |
| --- | --- |
|  | **[Design & Analysis of Algorithm]**  **[BSCS – 5 A]**  **Department of Computer Science**  **Bahria University, Lahore Campus** |

**Assignment: 3**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation of CLO** | **Question Number** | **Marks** | **Obtained Marks** |
| **CLO statement**   * **CLO: Demonstrate an understanding of algorithm design process and different problem solving techniques** | 1 | 2.5 |  |
| **Total Marks** | | **2.5** |  |

**Problem Title: Cafe Management [2.5 points]**

**Sweet Donuts, a new Coffee-and-donuts Café chain, wants to build cafes on many street corners of Lahore with the goal of maximizing their total profit. The street network is described as an undirected graph *G = (V, E)*, where the potential cafe sites are the vertices of the graph. Each vertex u has a nonnegative integer value *Pu*, which describes the potential profit of site u. Two cafes cannot be built on adjacent vertices (to avoid self-competition). You are supposed to design an algorithm that outputs the chosen set *U ⊆ V* of sites that maximizes the total profit ⅀u∈U Pu.**

**First, for parts (a)–(c), suppose that the street network G is acyclic, i.e., a tree.**

(a) [0.5 points] Consider the following “greedy” cafe-placement algorithm: Choose the highest-profit vertex u0 in the tree (breaking ties according to some order on vertex names) and put it into U. Remove u0 from further consideration, along with all of its neighbors in G. Repeat until no further vertices remain. Give a counterexample to show that this algorithm does not always give a cafe placement with the maximum profit.

(b) [0.9 points] Give an efficient algorithm to determine a placement with maximum profit.

(c) [0.6 points] Suppose that, in the absence of good market research, owner decides that all sites are equally good, so the goal is simply to design a cafe placement with the largest number of locations. Give a simple greedy algorithm for this case, and prove its correctness.

(d) [0.5 points] Now suppose that the graph is arbitrary, not necessarily acyclic. Give the fastest correct algorithm you can for solving the problem.