(CS1.301) Algorithm Analysis and Design (Monsoon 2022) Date: 27.08.2022

Deep Quiz 1

Alloted time: 45 minutes Total marks: 15

Roll number:

Room number:

Marks (To be filled by the evaluator)

Name of the evaluator	
Question 1	
Question 2	

Instructions:

- There are a total of 2 questions. Please answer the questions in the space provided.
- Discussions amongst the students are not allowed. No electronic devices nor notes/books of any kind are allowed.
- Any dishonesty shall be penalized heavily.
- Place your identity cards on the table for verification.
- Be clear in your arguments. Vague arguments shall not be given any credit.

Question 1 Suppose that an n node undirected graph $G = (V, E)$ contains two nodes s and t such that the distance between s and t is strictly greater than $n/3$. Show that there must exist at most two nodes u and v, each not equal to either s or t, such that deleting u and v from G destroys all s to t paths. Give an algorithm with running time $O(m + n)$ to find such nodes u and v. [5 marks]

Question 2 Suppose we are given a set U of objects labeled $p_1, p_2, \ldots, p_n$. For each pair p_i and p_j, we have a numerical distance $d(p_i, p_j)$. We futher have the property that for all $1 \leqslant i \leqslant n$, $d(p_i, p_i) = 0$ and for all $1 \leqslant i \neq j \leqslant n$, $d(p_j, p_i) = d(p_i, p_j) > 0$.

For a given parameter k as input, k-clustering of U is a partition of U into k nonempty sets $C_1, C_2, \ldots, C_k$. Spacing of a k-clustering is defined to be the minimum distance between any pair of points lying in different clusters. That is,

$$\text{Spacing}(C_1, C_2, \ldots, C_k) = \min_{1 \leqslant u \neq v \leqslant k} \left\{ \min\{d(p, p') \mid p \in C_u \text{ and } p' \in C_v\} \right\}.$$

Given that we want points in different clusters to be far apart from one another, a natural goal is to seek the k-clustering with the maximum possible spacing. In other words, we want to find the partition of U into k non-empty sets that maximizes the following expression.

$$\max_{U = C_1 \sqcup C_2 \sqcup \ldots \sqcup C_k} \{\text{Spacing}(C_1, C_2, \ldots, C_k)\}$$

The question now becomes the following – how can we efficiently find the one that has maximum spacing? [10 marks]

[Hint: It is related to one of the greedy algorithms we studied.]

Rough work