COL351: Analysis and Design of Algorithms
Instructor: Ragesh Jaiswal

1. You are given an array $A = A[1], A[2], ..., A[n]$ containing n integers and a positive integer k. Design an algorithm that outputs an array $C = C[1], C[2], ..., C[k]$ of size k such that

$$\sum_{i=1}^{n} \min_{j \in \{1,...,k\}} \{|A[i] - C[j]|\}$$

is minimized. Here $|x - y|$ denotes the absolute value of the difference between x and y. Discuss running time for your algorithm.

(*For example, if* $A = [0, 1, 2, 10, 11, 12]$ *and* $k = 2$*, then* $C = [1, 11]$)

2. **Another algorithm for max-flow**
Consider the following slightly changed version of the Ford-Fulkerson max-flow algorithm. This algorithm is also due to Jack Edmonds and Richard Karp.

> `Max-Flow(`G`)`
> - Start with a flow f such that $\forall e \in E, f(e) = 0$.
> - While there is an $s - t$ path in G_f
>   - Find an $s - t$ path P in G_f with **largest bottleneck value**.
>   - Augment along P to obtain f'.
>   - Update f to f' and G_f to $G_{f'}$
> - return(f).

(a) Think of an algorithm to find the *largest bottleneck path* from s to t in a given graph. A bottleneck path is a path such that the bottleneck edge has maximum weight. Discuss its running time.
(*Hint: Try ideas from Dijkstra's Algorithm.*)

(b) Let f be any $s - t$ flow and t be the value of maximum flow in the residual graph G_f. Let f' be the new flow after one augmentation and t' be the value of the new maximum flow in the residual graph $G_{f'}$. Argue that $t' \leq (1 - 1/m) \cdot t$.

(c) Use the properties you showed above to argue that for a graph with integer capacities, the algorithm runs in time $O(m^2 \cdot \log m \cdot \log f^*)$, where f^* is the value of the max-flow in the original graph G.