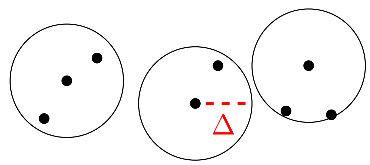
CSE421: Design and Analysis of Algorithms	May 5th, 2022
Homework 5	
Shayan Oveis Gharan	Due: May 12, 2022 at 23:59 PM

P1) We want to design an $O(1)$ approximation algorithm for the following clustering problem. Given a set of n points $p_1, \dots, p_n \in \mathbb{R}^d$, and an integer $1 \le k \le n$, find the minimum radius Δ and a set of balls of radius Δ centered at k of the given points such that all of the n points lie in these balls. Note that the balls have radius Δ with respect to the Euclidean distance.



a) (15 points) Assume that we know the optimum radius Δ. Design a polynomial time algorithm that finds at most k balls of radius $O(\Delta)$ centered at k of the points covering **all** of the given points.

Hint. Recall the triangle inequality: For any triple points $a, b, c \in \mathbb{R}^d$, $\|a - c\|_2 \le \|a - b\|_2 + \|b - c\|_2$.

b) (5 points) Now, assume that we do not know Δ. Instead suppose we know that the optimum Δ is in the interval $[1, R]$[1]. Use part (a) to design an algorithm that runs in time polynomial in $n, \log R$ to find the k balls of radius $O(\Delta)$.

P2) Draw the dynamic programming table of the following instance of the knapsack problem: You are given 5 items with weight $1, 3, 5, 7, 9$ and value $1, 2, 4, 5, 7$ respectively and the size of your knapsack is 14.

P3) You are a cashier at a Grocery store in a country with coins of values $v_1, v_2, \dots, v_n$ dollars (you can assume $v_1, \dots, v_n$ are positive integers). Furthermore, assume you have an infinite supply of each coin. A customer comes and you need to make a change for k dollars. Design an algorithm that runs in time polynomial in n, k and outputs the minimum number of coins you can use to make a change for k dollars.

P4) Suppose we have an infinite copies of an item that we want to sell. There are n buyers in the market where the i-th buyer has value v_i for this item. We run the process for k days; in day i we set a price of p_i and every buyer whose value for the item is *at least* p_i will buy the item, i.e., all j where $v_j \ge p_i$. Once a buyer buys the item he/she leaves the market and never buys again. Design an algorithm that runs in time polynomial polynomial in n and finds the best prices $p_1, \dots, p_k$ in order to maximize the total money we earn by day k. For example, if $k = 2$ and $v_1 = 2, v_2 = 5, v_3 = 4$ then the optimal solution is to set $p_1 = 4, p_2 = 2$. Then, in day 1, buyers $2, 3$ buy the item and we earn 8\$ and in day two buyer 1 buys the item and we earn 2\$. So, in total we receive 10\$.

[1] In practice you can take $1, R$ correspond to be the minimum/maximum pairwise of all of the given points

P5) **Extra Credit:** A k-hypergraph is composed of a set V of vertices and a set of hyperedges where every hyperedge is a subset of V of size at least 2 and at most k, i.e., S is a hyperedge if $S \subseteq V$ and $2 \leq |S| \leq k$. Note that 2-hypergraph is the same as a graph. Given a k-hypergraph $G = (V, E)$ with n vertices where for some $k \geq 2$ design a k-approximation algorithm for the vertex cover problem: Find the minimum set W of vertices of G such that every hyperedge $S \in E$ has at least one vertex of W.