COL 351 : Analysis and Design of Algorithms
Semester I, 2021-22, CSE, IIT Delhi

Assignment - 3 (due on 24th October, 11:00 PM)

Important Guidelines:

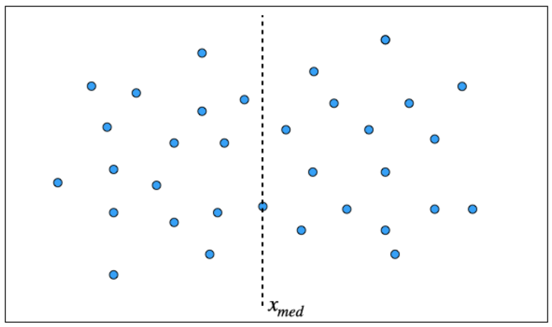
- Each assignment must be done in a group of size at most two.
- Handwritten submissions will not be accepted. Solutions must be typed-up (in Latex, Microsoft Word, etc.), and submitted in pdf format. Each solution must start on a new page.
- **Your answer to each question must be formal and have a proper correctness proof.** No marks will be granted for vague answers with intuition or for algorithms without proof. You must be very rigorous in providing mathematical detail in support of your arguments.
- Cheating of any form will lead to strict penalty.

1 Convex Hull

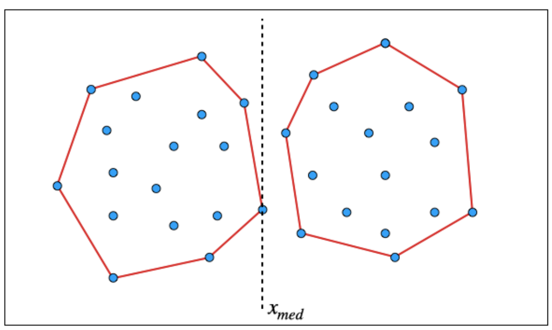
The Convex Hull of a set P of n points in x-y plane is a minimum subset Q of points in P such that all points in P can be generated by a convex combination of points in Q. In other words, the points in Q are *corners* of the convex-polygon of smallest area that encloses all the points in P.

Design an $O(n \log n)$ time Divide-and-Conquer algorithm to compute the convex hull of a set P of n input points $\{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n)\}$. [15 marks]

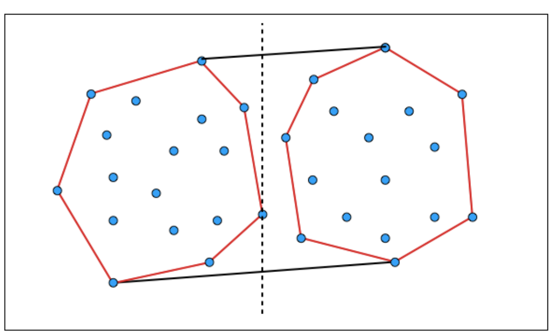
Hint:



Step 1: Split P into sets P_1 and P_2 of size $\lceil n/2 \rceil$ by diving along median of x-coordinate of all the points.



Step 2: Recursively compute convex hull of sets P_1 and P_2.



Step 3: Finally combine convex hull of P_1 and P_2 in linear time to obtain convex hull of P.

2 Particle Interaction

Some physicists are working on interactions among large numbers of very small charged particles. Basically, their set-up works as follows. They have an inert lattice structure, and they use this for placing charged particles at regular spacing along a straight line. Thus we can model their structure as consisting of the points $\{1, 2, 3, ..., n\}$ on the real line; and at each of these points j, they have a particle with charge q_j. (Each charge can be either positive or negative.)

They want to study the total force on each particle, by measuring it and then comparing it to a computational prediction. This computational part is where they need your help. The total net force on particle j, by Coulomb's Law, is equal to

$$F_j = \sum_{i<j} \frac{C\, q_i\, q_j}{(j-i)^2} - \sum_{i>j} \frac{C\, q_i\, q_j}{(j-i)^2}\ .$$

They have written the following simple program to compute F_j, for all j.

for $j = 1, 2, ..., n$ **do**
    Initialize F_j to 0;
    for $i = 1, 2, ..., n$ **do**
        if $i < j$ **then**
            Add $\frac{C\, q_i\, q_j}{(j-i)^2}$ to F_j;
        else if $i > j$ **then**
            Add $-\frac{C\, q_i\, q_j}{(j-i)^2}$ to F_j;
        end
    end
    Output F_j;
end

The running time of this program is $O(n^2)$. Your task is to design an algorithm that computes all the forces F_j in $O(n \log n)$ time. [15 marks]

3 Distance computation using Matrix Multiplication

Let $G = (V, E)$ be an unweighted undirected graph, and $H = (V, E_H)$ be a undirected graph obtained from G that satisfy: $(x, y) \in E_H$ if and only if $(x, y) \in E$ or there exists a $w \in V$ such that $(x, w), (w, y) \in E$. Further, let D_G denote the distance-matrix of G, and D_H be the distance-matrix of H.

(a) Prove that the graph $H = (V, E_H)$ can be computed from G in $O(n^\omega)$ time, where ω is the exponent of matrix-multiplication. [2 marks]

(b) Argue that for any $x, y \in V$, $D_H(x, y) = \left\lceil \frac{D_G(x, y)}{2} \right\rceil$. [2 marks]

(c) Let A_G be adjacency matrix of G, and $M = D_H * A_G$. Prove that for any $x, y \in V$, the following holds. [8 marks]

$$D_G(x,y) = \begin{cases} 2D_H(x,y) & M(x,y) \geq \text{degree}_G(y) \cdot D_H(x,y) \\ 2D_H(x,y) - 1 & M(x,y) \lneq \text{degree}_G(y) \cdot D_H(x,y) \end{cases}$$

(d) Use (c) to argue that D_G is computable from D_H in $O(n^\omega)$ time. [1 marks]

(e) Prove that all-pairs-distances in n-vertex unweighted undirected graph can be computed in $O(n^\omega \log n)$ time, if ω is larger than two. [2 marks]

4 Universal Hashing

Let $U = [0, M-1]$ be a universe of M elements, p be a prime number in range $[M, 2M]$, and $n(<< M)$ be an integer. Consider the following hash functions (where $r \in [1, p-1]$):

$$\begin{aligned} H(x) &:= (x) \mod n \\ H_r(x) &:= ((rx) \mod p) \mod n \end{aligned}$$

(a) Compute a random set S as follows:
− Intialize S to empty.
− Repeat n times: choose a random integer in U and add it to S.
Prove the following. [7 marks]

$$Prob(\text{max-chain-length in hash table of } S \text{ under hash-function } H(\cdot) > \log_2 n) \leqslant \frac{1}{n}$$

(b) Prove that for any given $r \in [1, p-1]$, there exists at least ${}^{M/n}C_n$ subsets of U of size n in which maximum chain length in hash-table corresponding to $H_r(x)$ is $\Theta(n)$. [3 marks]

(c) Implement $H()$ and $H_r()$ in Python/Java for $M = 10^4$ and the following different choices of sets of size $n = 100$: For $k \in [1, n]$, S_k is union of $\{0,\ n,\ 2n,\ 3n,\ \ldots,\ (k-1)n\}$ and $n - k$ random elements in U.

Obtain a plot of Max-chain-length for hash functions $H()$, $H_r()$ over different choices of sets S_k defined above. Note that you must choose a different random r for each choice of S_k. Provide a justification for your plots. [5 marks]