Randomized Computation

- Deterministic and (Las Vegas & Monte Carlo) Randomized Algorithms
- Probability Review
- Probabilistic Analysis of deterministic QUICK-SORT Algorithm
- RANDOMIZED-SELECT and RANDOMIZED-QUICK-SORT
- Max-Cut
- Min-Cut
- MAX-3-SAT and Derandomization
- Closest Pair
- Randomized Complexity Classes

IMDAD ULLAH KHAN

Closest Pair of Points Problem

Given n points in a plane, find a pair of points with minimum Euclidean distance between them

For $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

can be computed in $O(1)$

Applications: Computer graphics, computer vision, geographic information systems, molecular modeling, air traffic control

Brute force Algorithm:

FINDMIN among all $\binom{n}{2}$ pairwise distances $\triangleright$ $O(n^2)$ comparisons

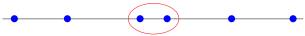
Closest Pair of Points Problem

Input: $P = \{p_1, p_2, \ldots, p_n\}$: a set of n distinct points in $\mathbb{R}^2$

Output: A pair of distinct points in P that minimizes the $d(p, q)$

1-dimensional space:

1. Sort points $\triangleright$ $O(n \log n)$
2. Find closest adjacent points $\triangleright$ $O(n)$



2-dimensional space:

- Divide and Conquer Algorithm $\triangleright$ $O(n \log n)$

Randomized Algorithm for Closest Pair

Input: $P = \{p_1, p_2, \ldots, p_n\}$: a set of n distinct points in $\mathbb{R}^2$

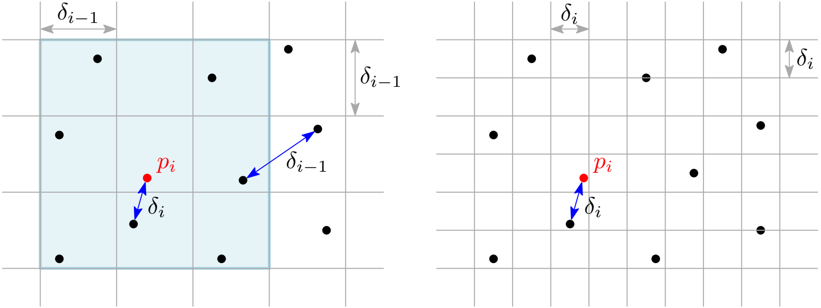
Output: A pair of distinct points in P that minimizes the $d(p, q)$

Assumptions

- All points are in the unit square $0 \leq x_i, y_i \leq 1$ $\triangleright$ WLOG
- Distance between each pair of points is distinct

A Randomized Incremental Algorithm

- Let $P = \{p_1, p_2, \cdots, p_n\}$ be a fixed random order
- $S_i = \{p_1, p_2, \cdots, p_i\}$: the first i points in P
- δ_i : the distance of the closest pair in S_i ▷ need δ_n
- Idea is to begin with S_2 lay out a grid G with cell size $\delta_2 \times \delta_2$
- For $i = 3$ to n insert point p_i in G incrementally
- In each step, update G cell size if $\delta_i < \delta_{i-1}$



A Randomized Incremental Algorithm: Implementation

- Given δ_{i-1}, how can we compute δ_i?
  - $\delta_i = \min d(p_i, p_j)\ \forall\ j$ in the neighborhood of p_i if $d(p_i, p_j) < \delta_{i-1}$

    $\triangleright$ Why? Distance between p_i and points outside adjacent cells of p_i is at least δ_{i-1} by construction
- What operations do we need for the grid structure?
  - BUILD-GRID(S, δ): build grid G with cell size δ & insert all points in S
  - INSERT-POINT(p_i): insert p_i
  - LOCATE-CELL(p_i): return cell containing p_i
  - GET-POINTS(c): return points in cell c
- Use hashing to implement grid so operations take $\mathcal{O}(1)$ time
  - Key universe is IDs of all cells in the grid
  - Actual key space is the IDs of cells containing points
  - Point co-ordinates are the data for each key
  - Cell containing p_i is located at in grid $\left(\lfloor x_i/\delta_{i-1} \rfloor, \lfloor y_i/\delta_{i-1} \rfloor\right)$

A Randomized Incremental Algorithm: Runtime

Algorithm Randomized Closest Pair: returns distance

function CLOSEST-PAIR(P)
  $\{p_1, p_2, \cdots, p_n\} \leftarrow$ RANDOM-PERMUTATION(P)
  $S_2 \leftarrow \{p_1, p_2\}$
  $G \leftarrow$ BUILD-GRID(S, δ_2)
  for $i = 3 \rightarrow n$ **do**
    $S_i \leftarrow S_{i-1} \cup p_i$ $\triangleright \mathcal{O}(1)$
    Compute δ_i $\triangleright \mathcal{O}(1)$
    if $\delta_i < \delta_{i-1}$ **then** G.BUILD-GRID(S, δ_i) $\triangleright \mathcal{O}(i)$
    else
      G.INSERT-POINT(p_i) $\triangleright \mathcal{O}(1)$
  return δ_n

A Randomized Incremental Algorithm: Runtime

- Given S_i, $\delta_i < \delta_{i-1}$ when $p_i \in C$ for any permutation of S_i

$$Pr[\delta_i < \delta_{i-1} | S_i] \;=\; \frac{2(i-1)!}{i!} \;=\; \frac{2}{i}$$

- The $\binom{n}{i}$ choices of S_i are equally likely $\implies \sum_{j \in \binom{n}{i}} Pr[S_{i_j}] = 1$

$$Pr[\delta_i < \delta_{i-1}] \;=\; \sum_{j \in \binom{n}{i}} Pr[\delta_i < \delta_{i-1} | S_{i_j}] \cdot Pr[S_{i_j}] \;=\; \frac{2}{i} \sum_{j \in \binom{n}{i}} Pr[S_{i_j}] \;=\; \frac{2}{i}$$

- Let X_i be the runtime of iteration i
- $E[X_i] = \mathcal{O}(1) + \mathcal{O}(i) \cdot Pr[\delta_i < \delta_{i-1}] = \mathcal{O}(1) + \mathcal{O}(i) \cdot {}^2/_i = \mathcal{O}(1)$

$$E[X] = \sum_{i=1}^{n} E[X_i] = \mathcal{O}(n)$$