Divided and Conquer

August 23, 2025

Closest Pair of Points

Input

Points $P_1, \ldots, P_n \in \mathbb{R}^2$. Each point is represented by $P_i = (x_i, y_i) \in \mathbb{R}^2$

Output

The closest pair P_i, P_j.

An Example

A Brute Force Algorithm

Closest Pair in a Line

Closest Pair in Two Line

Divided and Conquer

Partition Space into Grids

Dividend and Conquer Algorithm

```
Closest-Pair(P)
  Construct P_x and P_y   (O(n log n) time)
  (p*_0, p*_1) = Closest-Pair-Rec(P_x, P_y)

Closest-Pair-Rec(P_x, P_y)
  If |P| ≤ 3 then
    find closest pair by measuring all pairwise distances
  Endif

  Construct Q_x, Q_y, R_x, R_y (O(n) time)
  (q*_0, q*_1) = Closest-Pair-Rec(Q_x, Q_y)
  (r*_0, r*_1) = Closest-Pair-Rec(R_x, R_y)

  δ = min(d(q*_0, q*_1), d(r*_0, r*_1))
  x* = maximum x-coordinate of a point in set Q
  L = {(x,y) : x = x*}
  S = points in P within distance δ of L.

  Construct S_y (O(n) time)
  For each point s ∈ S_y, compute distance from s
     to each of next 15 points in S_y
     Let s, s' be pair achieving minimum of these distances
     (O(n) time)

  If d(s,s') < δ then
     Return (s,s')
  Else if d(q*_0, q*_1) < d(r*_0, r*_1) then
     Return (q*_0, q*_1)
```

Thanks!