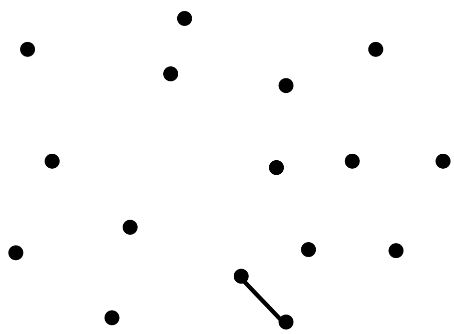
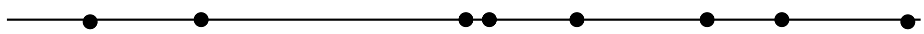
Closest Pair Problem

- **Given:** n points in the plane.
- **Find:** closest pair.



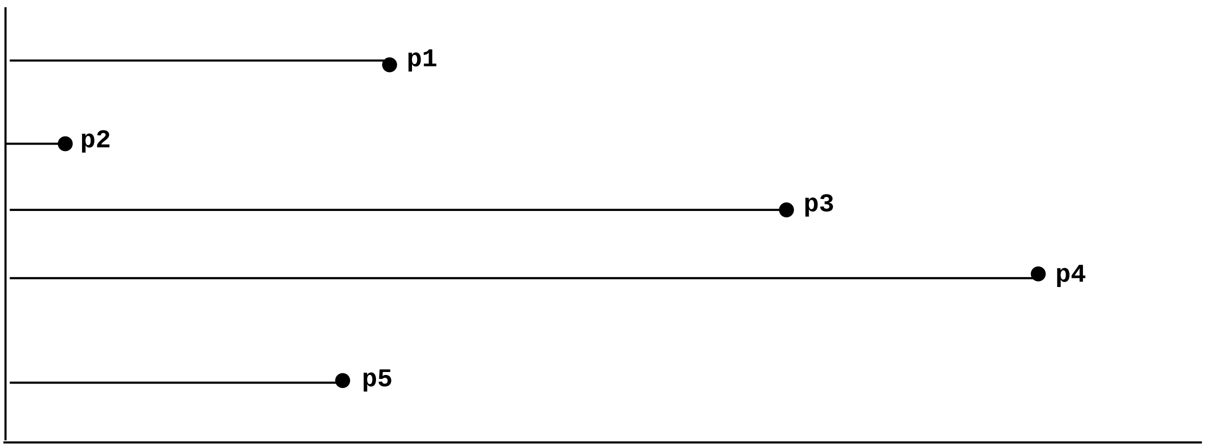
- Trivial algorithm $O(n^2)$
- Can it be improved?
- Yes, in 1 dimension.



- Sort. Closest pair is next to each other.
- Sorting $O(n \log n)$. Scanning $O(n)$ time.

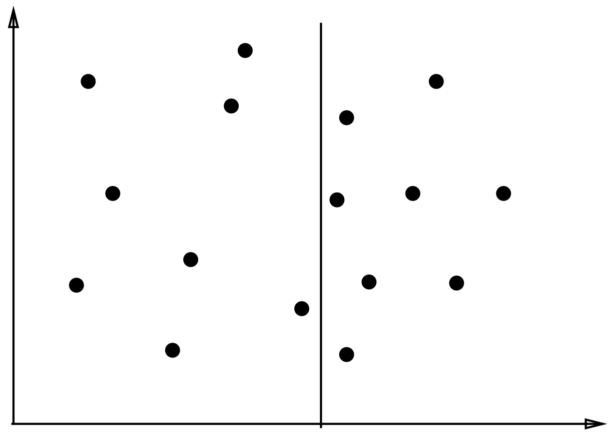
Closest Pair Problem

- Sorting provides an optimal $\Theta(n \log n)$ algorithm in 1-dimensional space.
- Can this be generalized to higher dimensions?
- Project onto one of the axes and then sort.



- Does not work. p_1 and p_5 are nearest neighbors but their projections are farthest away on the y-axis.

Closest Pair Problem - Divide and Conquer



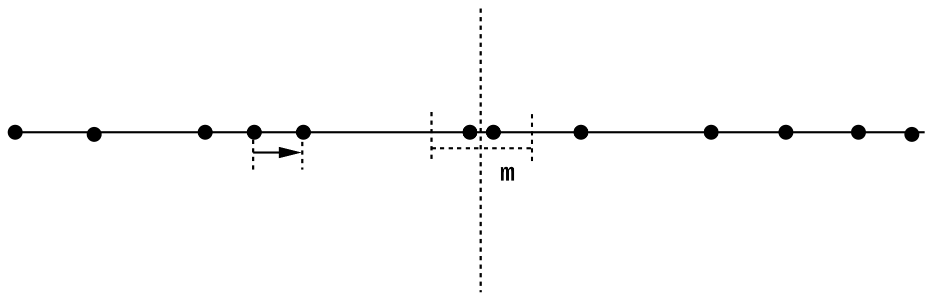
- Nearest neighbors in S_1.
- Nearest neighbors in S_2.
- Nearest neighbors, one in S_1 other in S_2.
- Time complexity:

$$T(n) = 2T(n/2) + O(n^2/4)$$

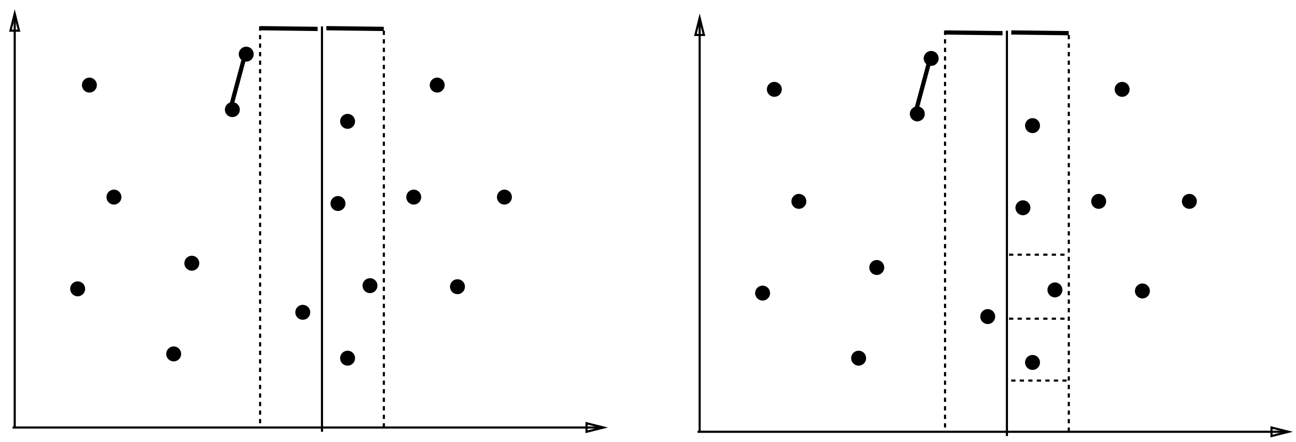
is $O(n^2)$

Closest Pair Problem - Divide and Conquer

- Is it necessary to check all $n^2/4$ pairs with one point in S_1 and the other point in S_2?
- In 1-dimensional space.

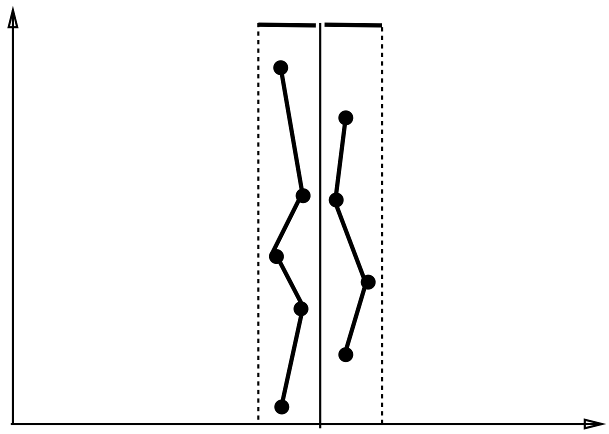


- Let $\sigma = \min\{|p_i p_j|, |q_k q_l|\}$
- Only points at distance σ need to be checked.
- There is at most one point in S_1 at distance σ from m. Similarly for S_2.
- In 2-dimensional space.



Closest Pair Problem - Divide and Conquer

- Preprocessing: Sort S by y-coordinates.
- Divide S into two equal size subsets S_1 and S_2 by a vertical median.
- Solve (recursively) for S_1 and S_2. Let $\delta = \min\{\delta_1, \delta_2\}$ where δ_i is the smallest distance in S_i, $i = 1, 2$.
- Determine the upward chain P_i through points of S_i at distance δ from the median. Can be done in $O(n)$ time.



- In total $\Theta(n \log n)$.
- This method cannot be generalized to solve other problems.