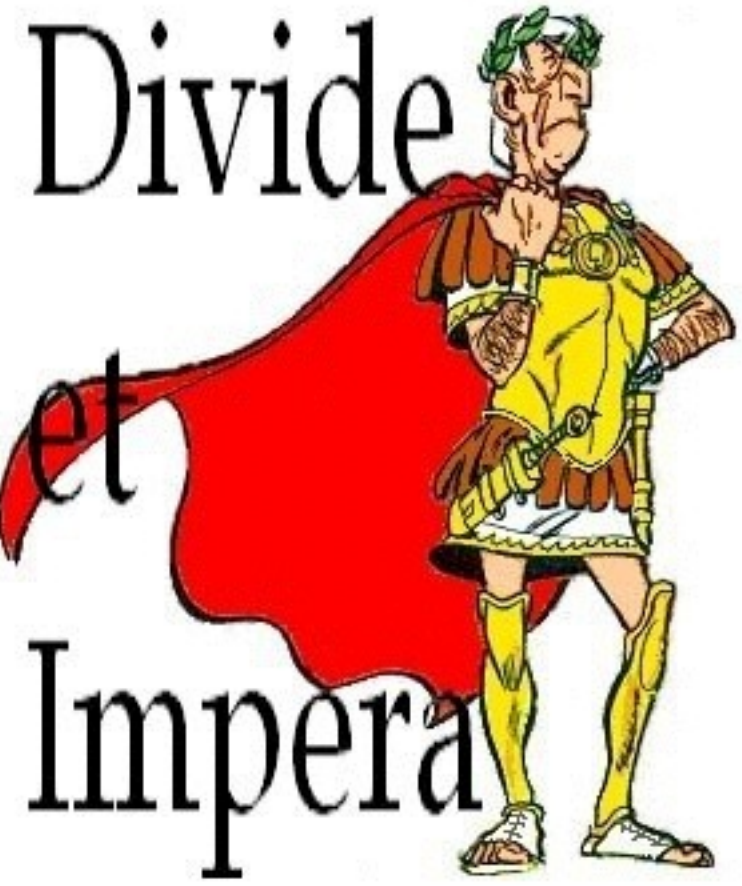


Divide & Conquer

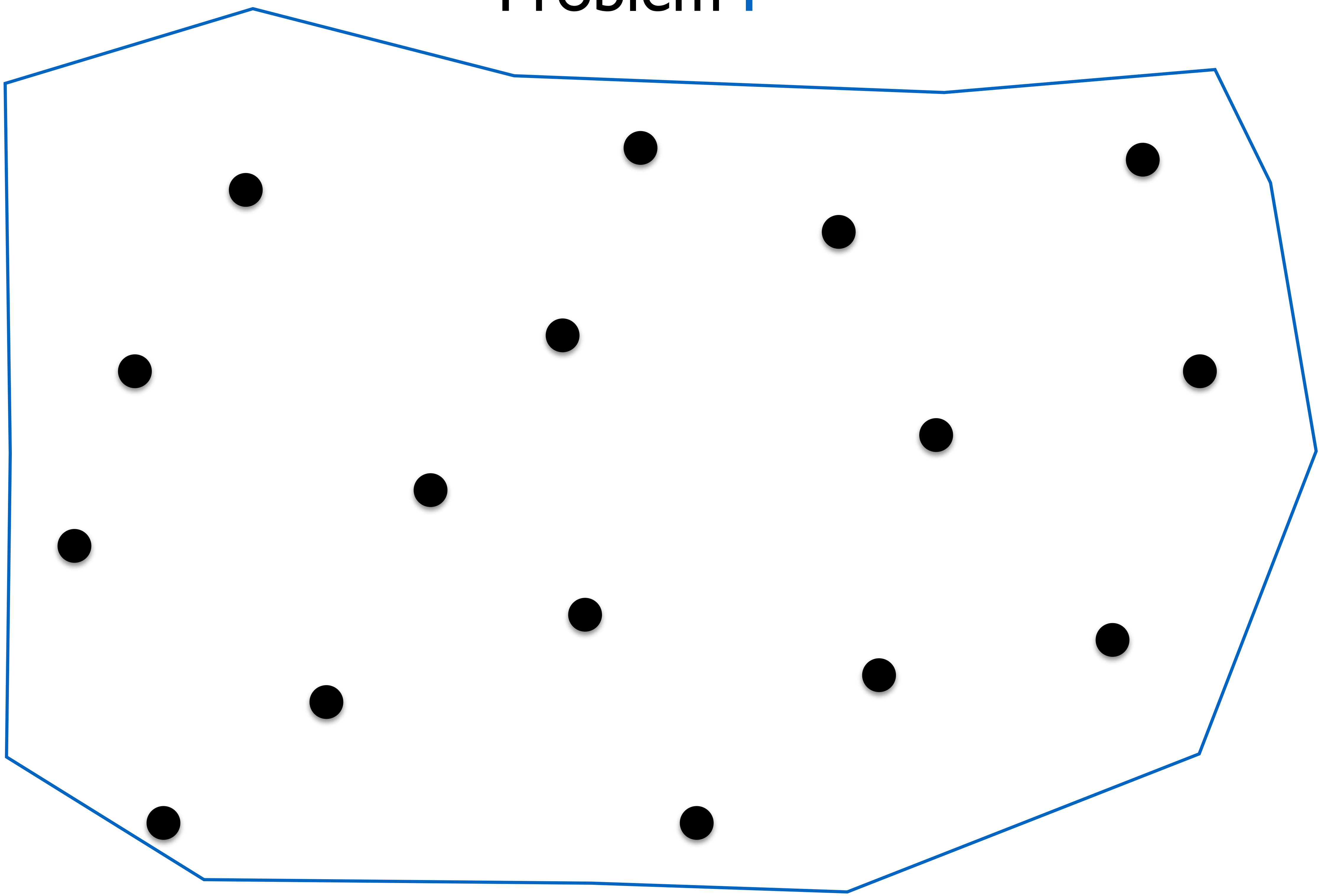
- Divide problem into subproblems of smaller size
- Solve subproblems recursively
- Combine solutions

Closest Pair

- Given *n* points in the plane find two whose distance is smallest!
- Naive solution: try all pairs in $O(n^2)$
- Can we do better?

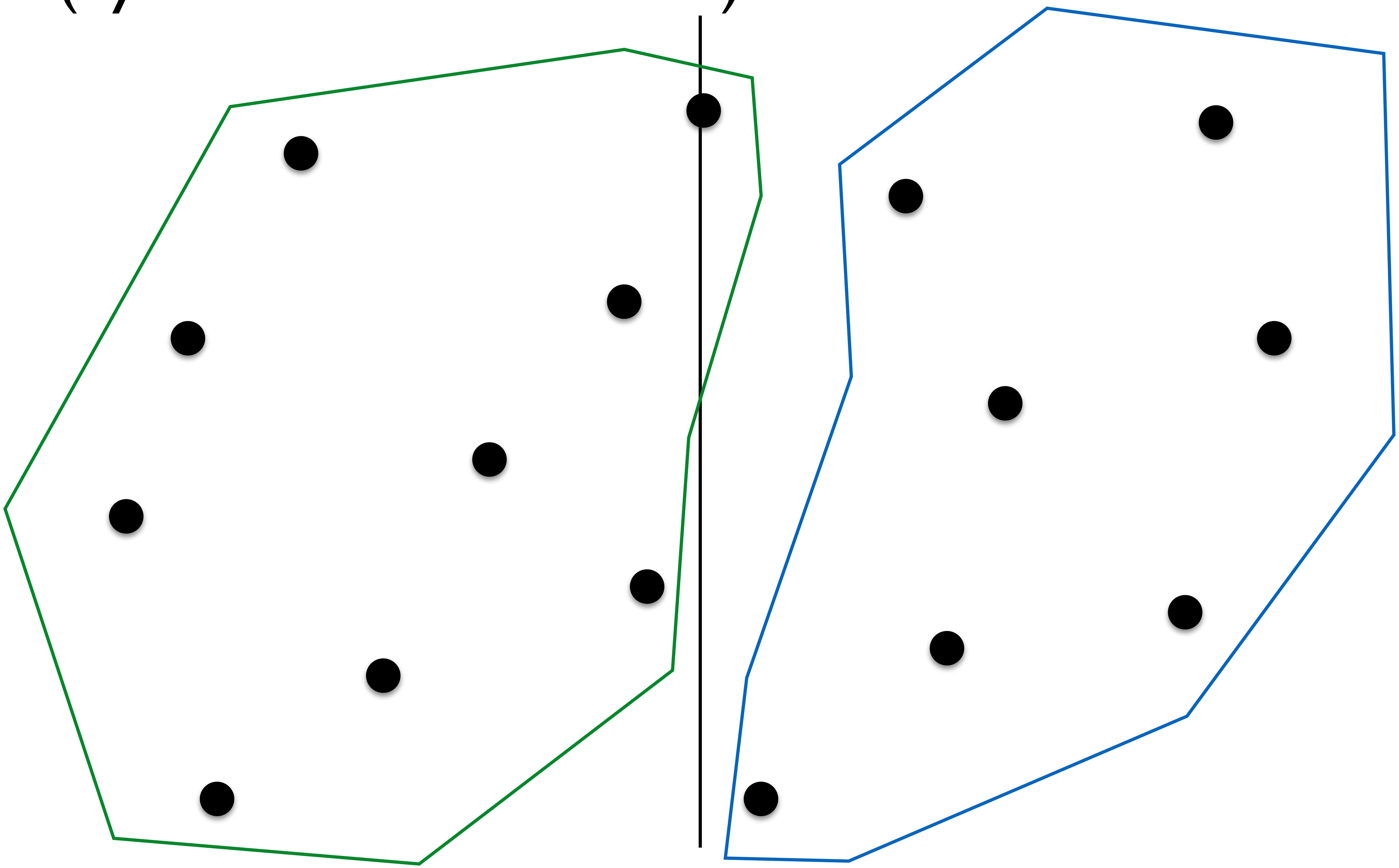
Closest Pair

Problem P



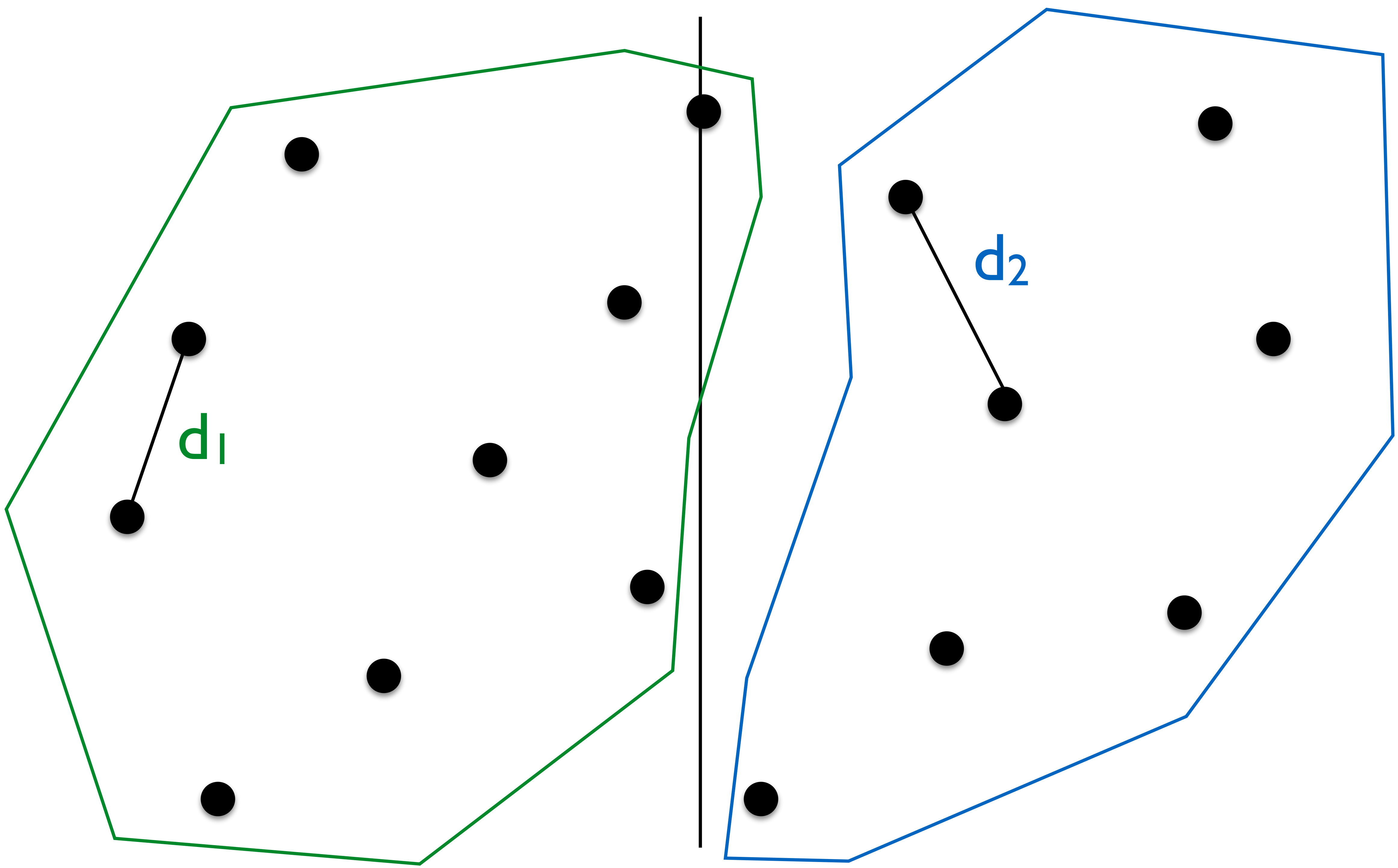
Closest Pair

Partition problem into subproblems P_1 and P_2 (by median x-coordinate)



Closest Pair

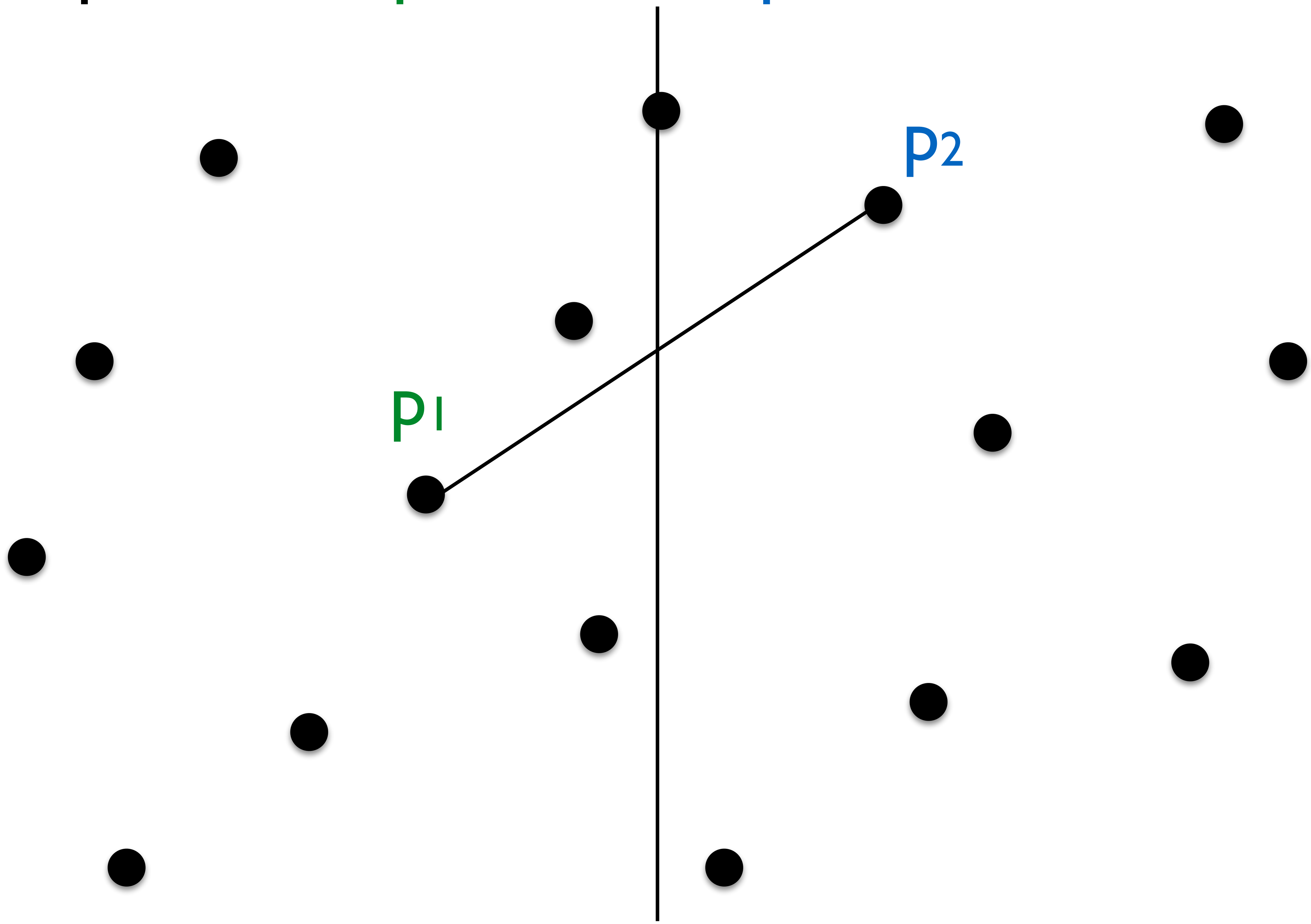
Recursively compute d_1 in P_1 and d_2 in P_2



Closest Pair

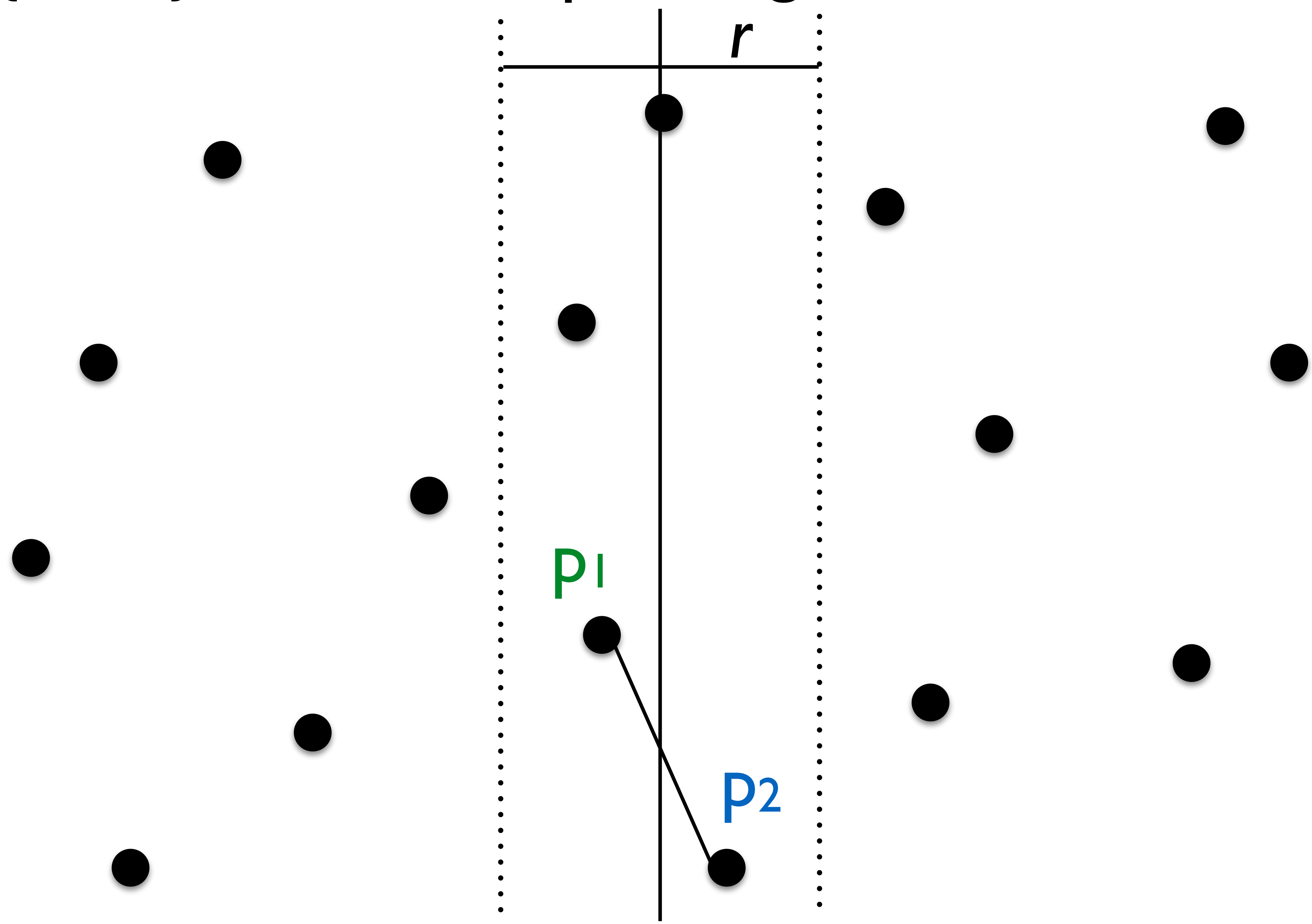
How can we combine the solutions?

Consider pairs with p_1 in P_1 and p_2 in P_2



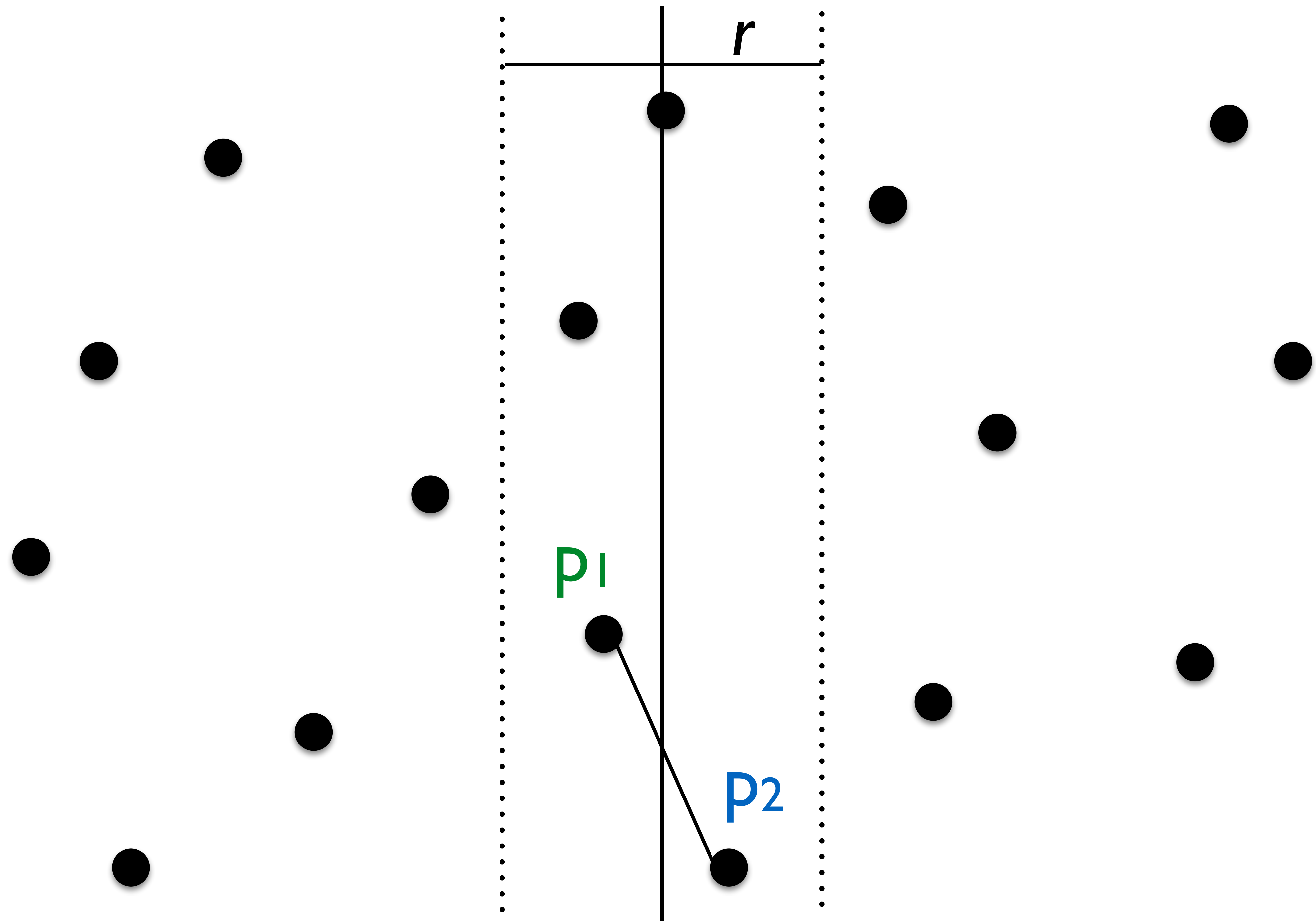
Closest Pair

We only need to consider points within distance $r = min\{d_1, d_2\}$ from the separating line



Closest Pair

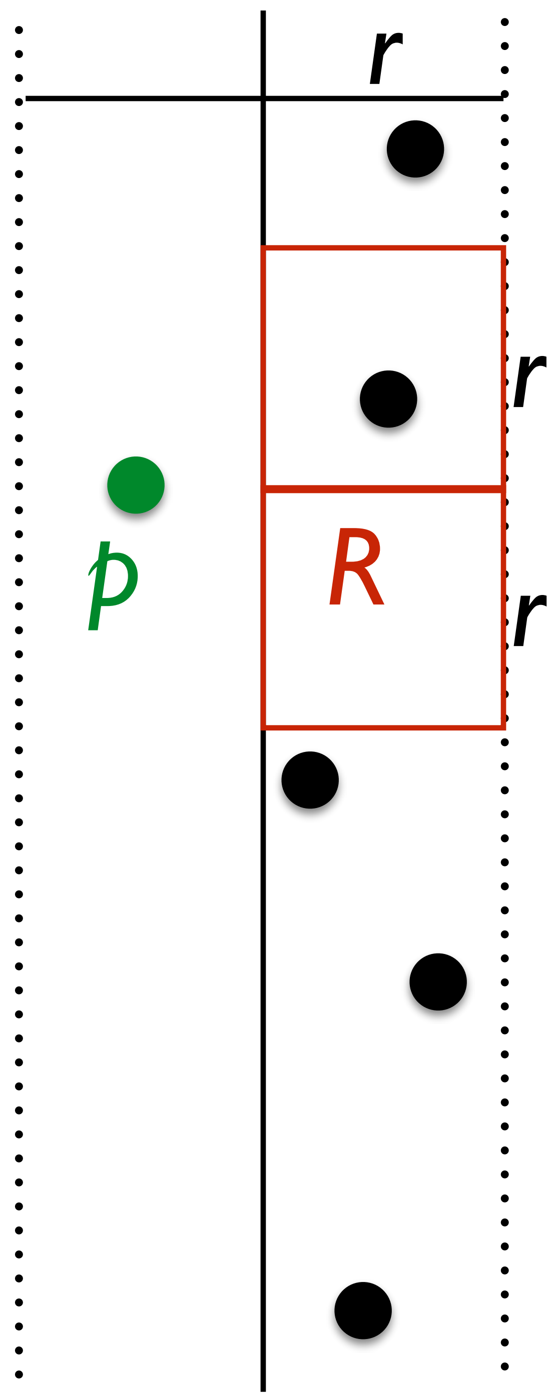
This could be $n^2/4$ pairs! :(



Closest Pair

We can improve!

Consider point in P_1

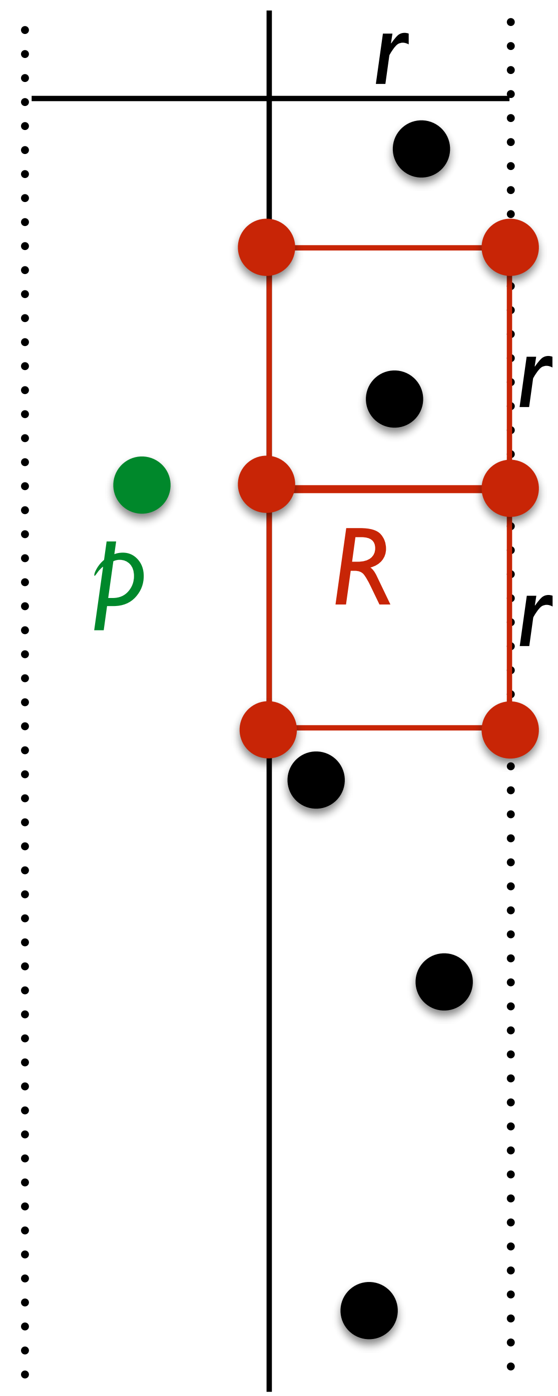


All points of P_2 within distance r must be in a r x $2r$ rectangle R

Closest Pair

We can improve!

Consider point in P_1



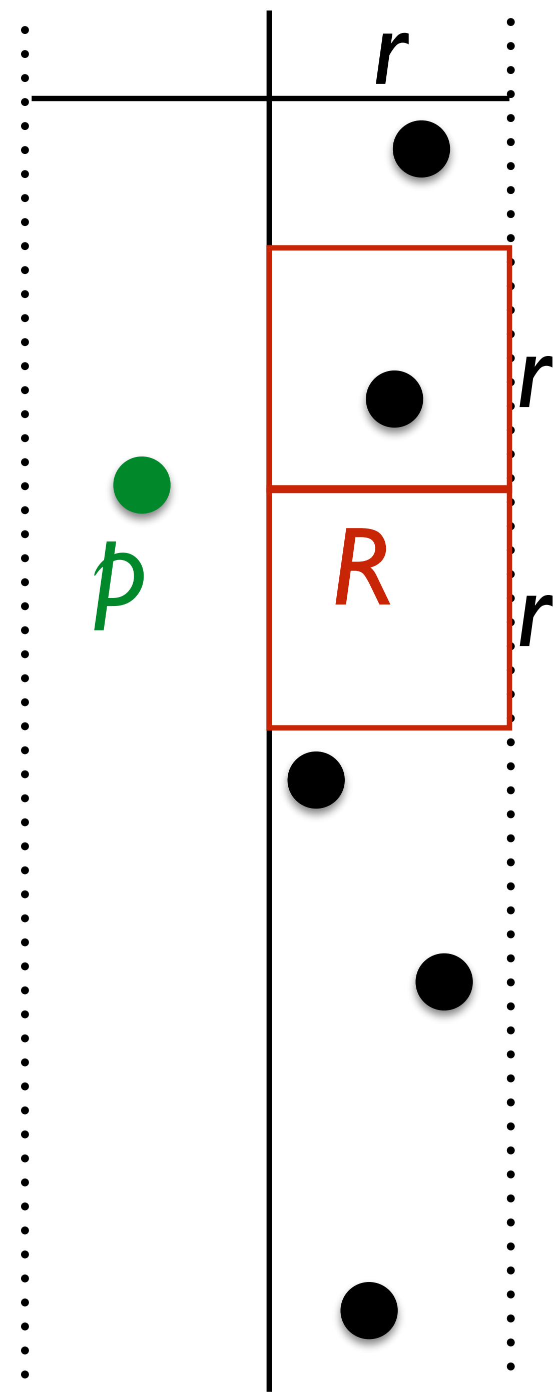
All points of P_2 within distance r must be in a r x $2r$ rectangle R

Since all points in P_2 have distance at least r, there can be at most 6 points in R

Closest Pair

We can improve!

Consider point in P_1



If the points of P_1 and P_2 are sorted by y-coordinate, we can do this for all p in $O(n)$ time

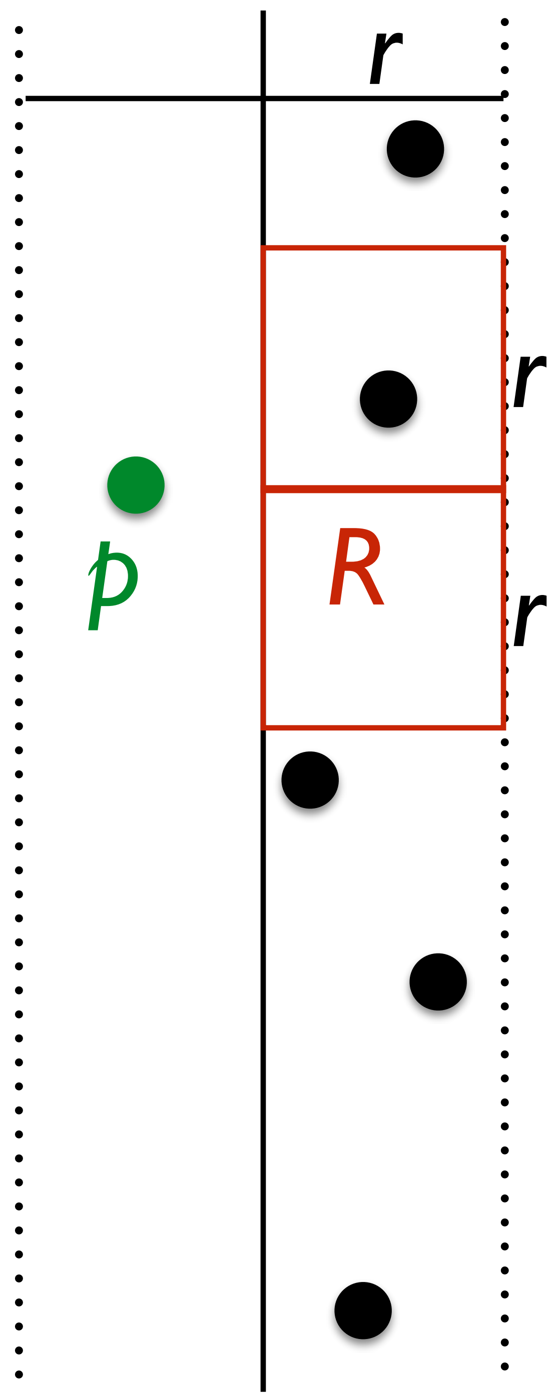
All points of P_2 within distance r must be in a r x $2r$ rectangle R

Since all points in P_2 have distance at least r, there can be at most 6 points in R

Closest Pair

We can improve!

Consider point in P_1



All points of P_2 within distance r must be in a r x $2r$ rectangle R

Since all points in P_2 have distance at least r, there can be at most 6 points in R

If the points of P_1 and P_2 are sorted by y-coordinate, we can do this for all p in $O(n)$ time

$O(nlogn)$