The closest pair point

Problem statement:

Given a set of n points on a line (1-dimensional, unsorted), find two points whose distance is shortest.

The closest pair point

Problem statement:

Given a set of n points on a line (1-dimensional, unsorted), find two points whose distance is shortest.

Remark:

- The problem is known as *the closest pair problem in 1-dimension*. Section 33.4 provides an algorithm for finding the closest pair of points in 2-dimension, i.e., on a plane, by extending the DC strategy we study here.

The closest pair point

A **brute-force** solution

- Pick two of n points and compute the distance

The closest pair point

A **brute-force** solution

- Pick two of n points and compute the distance

Cost:

$$T(n) = \binom{n}{2} = \frac{n!}{2!(n-2)!} = \Theta(n^2).$$

The closest pair point

Algorithm 1

1. Sort the points, say Merge Sort
2. Perform a linear scan

The closest pair point

Algorithm 1

1. Sort the points, say Merge Sort
2. Perform a linear scan

Remarks:

- Cost: $\Theta(n \lg n) + \Theta(n) = \Theta(n \lg n)$

The closest pair point

Algorithm 1

1. Sort the points, say Merge Sort
2. Perform a linear scan

Remarks:

- Cost: $\Theta(n \lg n) + \Theta(n) = \Theta(n \lg n)$
- Unfortunately, the algorithm cannot be extended to the 2-dimension case.

The closest pair point

Algorithm 2 (Divide-and-Conquer):

1. Divide the set S of n points by some point $mid \in S$ into two sets S_1 and S_2 such that

$$p < q \quad \text{for all } p \in S_1 \text{ and } q \in S_2$$

For example, $mid \in S$ can be the median, found in $O(n)$.

The closest pair point

Algorithm 2 (Divide-and-Conquer):

1. Divide the set S of n points by some point $mid \in S$ into two sets S_1 and S_2 such that

$$p < q \quad \text{for all } p \in S_1 \text{ and } q \in S_2$$

   For example, $mid \in S$ can be the median, found in $O(n)$.

2. Conquer:
   (a) finds the closest pair *recursively* on S_1 and S_2, gives us two closest pairs of points

$$\{p_1, p_2\} \in S_1 \text{ and } \{q_1, q_2\} \in S_2$$

The closest pair point

Algorithm 2 (Divide-and-Conquer):

1. Divide the set S of n points by some point $mid \in S$ into two sets S_1 and S_2 such that

$$p < q \quad \text{for all } p \in S_1 \text{ and } q \in S_2$$

   For example, $mid \in S$ can be the median, found in $O(n)$.

2. Conquer:

   (a) finds the closest pair *recursively* on S_1 and S_2, gives us two closest pairs of points

$$\{p_1, p_2\} \in S_1 \text{ and } \{q_1, q_2\} \in S_2$$

   (b) finds the *closest crossing pair* $\{p_3, q_3\}$ with $p_3 \in S_1$ and $q_3 \in S_2$.

The closest pair point

Algorithm 2 (Divide-and-Conquer):

1. Divide the set S of n points by some point $mid \in S$ into two sets S_1 and S_2 such that

$$p < q \quad \text{for all } p \in S_1 \text{ and } q \in S_2$$

   For example, $mid \in S$ can be the median, found in $O(n)$.

2. Conquer:
   (a) finds the closest pair *recursively* on S_1 and S_2, gives us two closest pairs of points

$$\{p_1, p_2\} \in S_1 \text{ and } \{q_1, q_2\} \in S_2$$

   (b) finds the *closest crossing pair* $\{p_3, q_3\}$ with $p_3 \in S_1$ and $q_3 \in S_2$.

3. Combine: the closest pair in the set S is

$$\text{argmin}\{|p_1 - p_2|, |q_1 - q_2|, |p_3 - q_3|\}.$$

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.
2. How many points of S_1 can lie in $(mid - d, mid]$?

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.
2. How many points of S_1 can lie in $(mid - d, mid]$?
   answer: at most one

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.
2. How many points of S_1 can lie in $(mid - d, mid]$?
   answer: at most one
3. How many points of S_2 can lie in $[mid, mid + d)$?

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.
2. How many points of S_1 can lie in $(mid - d, mid]$?
   answer: at most one
3. How many points of S_2 can lie in $[mid, mid + d)$?
   answer: at most one

The closest pair point

Remarks:

1. Both p_3 and q_3 must be within distance $d = \min\{|p_1 - p_2|, |q_1 - q_2|\}$ of mid if $\{p_3, q_3\}$ is to have a distance smaller than d.
2. How many points of S_1 can lie in $(mid - d, mid]$?
   answer: at most one
3. How many points of S_2 can lie in $[mid, mid + d)$?
   answer: at most one
4. Therefore, the number of pairwise comparisons that must be made between points in different subsets is thus at most one.

The closest pair point

```
ClosestPair(S)
if |S| = 2, then
   d = |S[2] - S[1]|
else
   if |S| = 1
      d = infty
   else
      mid = median(S)
      construct S1 and S2 from mid
      d1 = ClosestPair(S1)
      d2 = ClosestPair(S2)
      p3 = max(S1)
      q3 = min(S2)
      d = min(d1, d2, q3-p3)
   end if
end if
return d
```

The closest pair point

Remark:

1. A median of a set A is the "halfway point" of the set A can be found in linear time $\Theta(n)$ on average (see Chapter 9).

The closest pair point

Remark:

1. A median of a set A is the "halfway point" of the set A can be found in linear time $\Theta(n)$ on average (see Chapter 9).
2. The points in the intervals $(mid - d, mid]$ and $[mid, mid + d)$ can be found in linear time $O(n)$, called *linear scan*.

The closest pair point

Remark:

1. A median of a set A is the "halfway point" of the set A can be found in linear time $\Theta(n)$ on average (see Chapter 9).
2. The points in the intervals $(mid - d, mid]$ and $[mid, mid + d)$ can be found in linear time $O(n)$, called *linear scan*.
3. Total cost:
$$T(n) = 2 \cdot T(\frac{n}{2}) + \Theta(n) = \Theta(n \lg n).$$

The closest pair point

Remark:

1. A median of a set A is the "halfway point" of the set A can be found in linear time $\Theta(n)$ on average (see Chapter 9).
2. The points in the intervals $(mid - d, mid]$ and $[mid, mid + d)$ can be found in linear time $O(n)$, called *linear scan*.
3. Total cost:
$$T(n) = 2 \cdot T(\frac{n}{2}) + \Theta(n) = \Theta(n \lg n).$$
4. In general, given n points in d-dimension, the closest pair of points can be found in $O(n(\lg n)^{d-1})$.

Extra: Medians and order statistics

- Selection problem:
  Input:

  > *A set* A *of* n *(distinct) numbers and an integer* i, *with* $1 \leq i \leq n$.

Extra: Medians and order statistics

- Selection problem:
  Input:

  A set A of n (distinct) numbers and an integer i, with $1 \leq i \leq n$.

  Output:

  The element $x \in A$ that is larger than exactly $i - 1$ other elements of A. In other words, x is the ith smallest element of A.

Extra: Medians and order statistics

- Selection problem:
  Input:

  A set A of n (distinct) numbers and an integer i, with $1 \leq i \leq n$.

  Output:

  The element $x \in A$ that is larger than exactly $i - 1$ other elements of A. In other words, x is the ith smallest element of A.

- A median is the "halfway point" of the set A, i.e, $i = \lceil (n+1)/2 \rceil$.

Extra: Medians and order statistics

- Selection problem:
  Input:

  A set A of n (distinct) numbers and an integer i, with $1 \leq i \leq n$.

  Output:

  The element $x \in A$ that is larger than exactly $i - 1$ other elements of A. In other words, x is the ith smallest element of A.

- A median is the "halfway point" of the set A, i.e, $i = \lceil (n+1)/2 \rceil$.
- A simple sorting algorithm will take $O(n \lg n)$ time.

Extra: Medians and order statistics

- Selection problem:
  Input:

  A set A of n (distinct) numbers and an integer i, with $1 \leq i \leq n$.

  Output:

  The element $x \in A$ that is larger than exactly $i - 1$ other elements of A. In other words, x is the ith smallest element of A.

- A median is the "halfway point" of the set A, i.e, $i = \lceil (n+1)/2 \rceil$.
- A simple sorting algorithm will take $O(n \lg n)$ time.
- Yet, a DC strategy leads to running time of $O(n)$ — see Chapter 9.