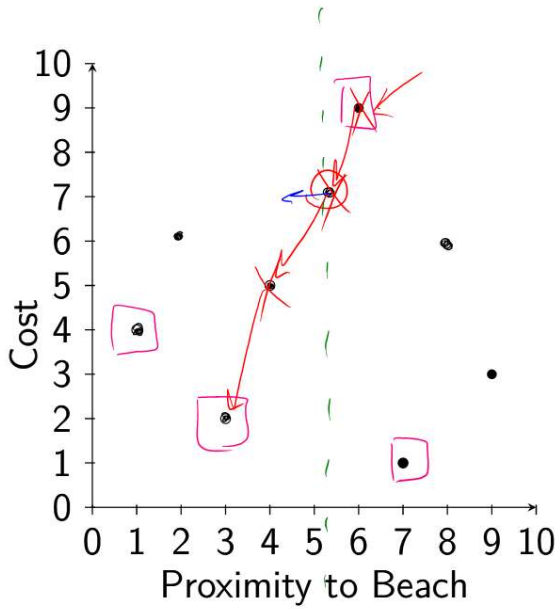
CompSci 161
Winter 2023 Lecture 07:
Divide and Conquer IV:
The Minima Set Problem

Minima Set Problem Statement

- We have a database of hotels.
- Each hotel has:
  - a proximity to the beach (x-coordinate)
  - a nightly room cost (y-coordinate)
  - Assume all coordinates distinct
- Want cheapest hotel closest to the beach
  - Might not be a unique hotel.
  - One might be closest, another cheapest.
  - Return **the set** that aren't wrong.
    - Any where no other hotel is both cheaper and closer.

3

Minima Set Example



4

Minima Set Brute Force

Sort hotels along any dimension
for $i = 1 \rightarrow n - 1$ **do**
  for $j = i + 1 \rightarrow n$ **do**
    if A_i is cheaper and closer than A_j **then**
      Remove A_j
return All hotels that we did not remove

- This is $O(n^2)$.

Polynomial time

5 Beginning Divide and Conquer

```
MinimaSet(S)
  if n ≤ 1 then
    return S
  p ← median point in S by x-coordinate
  L ← points less than p
  G ← points greater than or equal to p
  M1 ← MinimaSet(L)
  M2 ← MinimaSet(G)
```

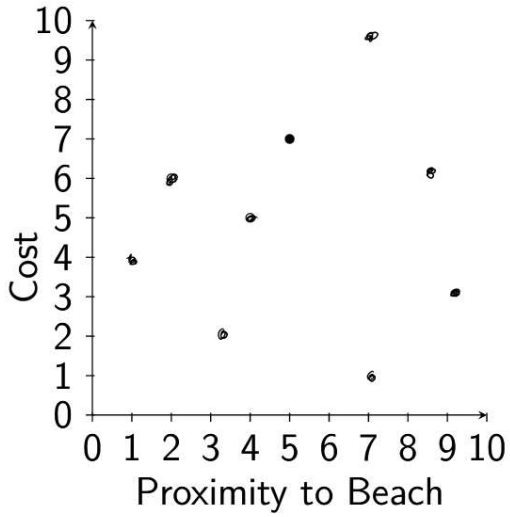
Quick Select expected $O(n)$ time

- return $M_1 \cup M_2$?

Every in M_1 in correct answer?

Yes...

6 Example revisited



- From $M_1 \cup M_2$, which point(s) belong for sure?

7

Finding a correct recombine

Pre-process w/ sort by x // $\Theta(n \log n)$ once

```
MinimaSet(S)
  if n ≤ 1 then
    return S
  p = median point in S by x-coordinate
  L = points less than p
  G = points greater than or equal to p
  M1 = MinimaSet(L)
  M2 = MinimaSet(G)
```

create together, keep in order

- return $M_1 \cup M_2$?
- How do I recombine correctly?

8

Improved Recombine

$T(n) = 2T(\frac{n}{2}) + n^2$ is $\Theta(n^2)$

```
M1 = MinimaSet(L)
M2 = MinimaSet(G)
for each a ∈ M1 do
  for each b ∈ M2 do
    if a better than b then
      remove b from M2
```

none of these can be eliminated

is a cheaper than b? Already closer to ocean.

- How can we improve the "recombine" step?
- What is the resulting running time?

mult: $T(n) = 4T(n/2) + n$ → 3

// All in M_2 further from beach than any in M_1

// So use least expensive cost in M_1

```
c = least cost in M1 (of any hotel)
for each b ∈ M2
    if b.cost > c
        remove b from M2
return M1 ∪ M2
```

$\Theta(n)$

$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$ is $\Theta(n \lg n)$