CompSci 161
Spring 2021 Lecture 09:
Divide and Conquer IV:
The Minima Set Problem

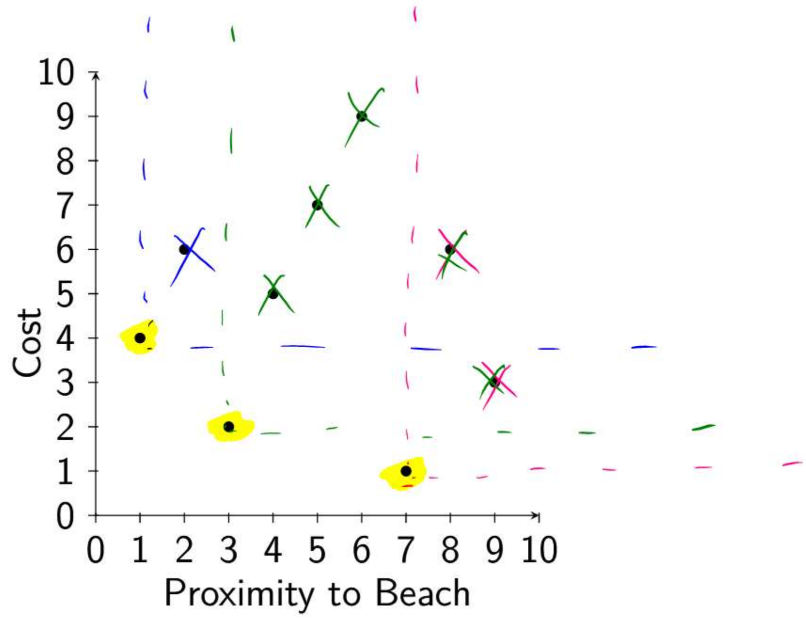
2

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)
- 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 $i = 1 \to n-1$ **do**
  for $j = i+1 \to n$ **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 $O(n^2)$.

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)
```

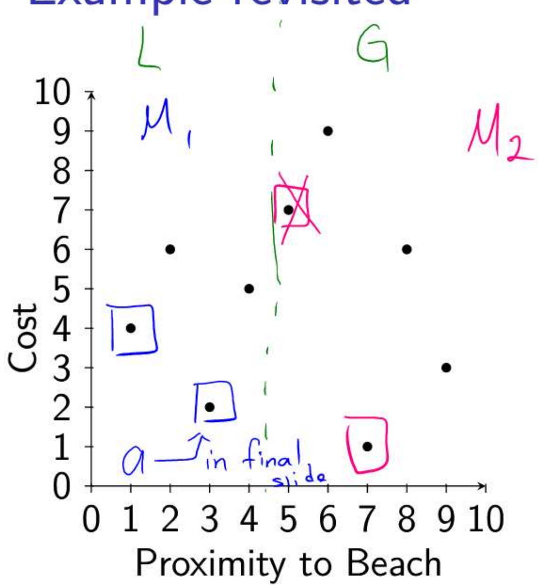
or brute force if $n \leq$ some constant, now $O(1)$

divide

recurse

- return $M_1 \cup M_2$?

6 # Example revisited



Observation: any element of M_2 eliminated by any $x \in M_1$, can also be eliminated by the cheapest element in M_1.

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

7 Finding a correct recombine

or $\Theta(n\lg n)$ once at start

```
MinimaSet(S)
  if n ≤ 1 then
    return S
  p ← median point in S by x-coordinate      ← Θ(n)
  L ← points less than p                      } Θ(n)
  G ← points greater than or equal to p       }
  M1 ← MinimaSet(L)                           } 2T(n/2)
  M2 ← MinimaSet(G)                           }
```

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

$O(n^2)$

```
for each a ∈ M1   // n/2 iterations
  for each b ∈ M2   // n/2 iterations
    if a eliminates b
      remove b from M2
return M1 ∪ M2
```

improve

$T(n) = 2T\left(\frac{n}{2}\right) + n^2$

8 Improved Recombine

Previous Code ↑ $\}\ \Theta(n)$

```
M1 ← MinimaSet(L)          } 2T(n/2)
M2 ← MinimaSet(G)          }
~~for each a ∈ M1 do~~   a ← cheapest in M1 // Θ(n)
for each b ∈ M2 do                         }
  if a better than b then                  } Θ(n) (only one
    remove b from M2                       }   value of a used)
```

- How can we improve the "recombine" step?

$T(n) = 2T(n/2) + n$ is $\Theta(n \log n)$