CompSci 260P
Spring 2023 Lecture 16: Divide and Conquer III

Integer Multiplication

- Given two n-bit integers X, Y, compute $X \cdot Y$
- Example: What is $13 \cdot 11$?

What is a Computer Anyway?

- Example: What is $13 \cdot 11$?

Al-Khwarizmi's algorithm

Why does al-Khwarizmi's algorithm work?

- Example: What is $13 \cdot 11$?

ALG on next page:

$X_L = 1111 = 15$

$X = 143 = \underbrace{1000}\,1111$

$X_H = 1000 = 8$

$143 = 8 \cdot 2^{8/2} + 15 = 8 \cdot 16 + 15$

Starting D&C for Integer Multiplication

$$X \times Y = (X_H \times 2^{n/2} + X_L) \times (Y_H \times 2^{n/2} + Y_L)$$
$$= X_H \cdot Y_H \times 2^n + (X_H Y_L + X_L Y_H) \times 2^{n/2} + X_L Y_L$$

Algorithm Mult(X, Y)

Create X_H, X_L, Y_H, Y_L

A = Mult(X_H, Y_H)

B = Mult(X_H, Y_L)

~~C = mult(X_L, Y_H)~~

~~D = mult(X_L, Y_L)~~

E = Mult($X_H + X_L$, $Y_H + Y_L$)

return $A \cdot 2^n + \cancel{(B+C)} \cdot 2^{n/2} + D$

$(E - A - D)$

$T(n) = \cancel{4}\,3\,T(\frac{n}{2}) + \Theta(n)$

$\cancel{= \Theta(n^2)}$

$\Theta(n^{\log_2 3})$

Improved Integer Multiplication

- Do we really need to compute $X_H \cdot Y_L$?

$$(X_H + X_L)(Y_H + Y_L)$$

$$E = \underbrace{X_H Y_H}_{A} + X_L Y_H + X_H Y_L + \underbrace{X_L Y_L}_{D}$$

Integer Multiplication Example

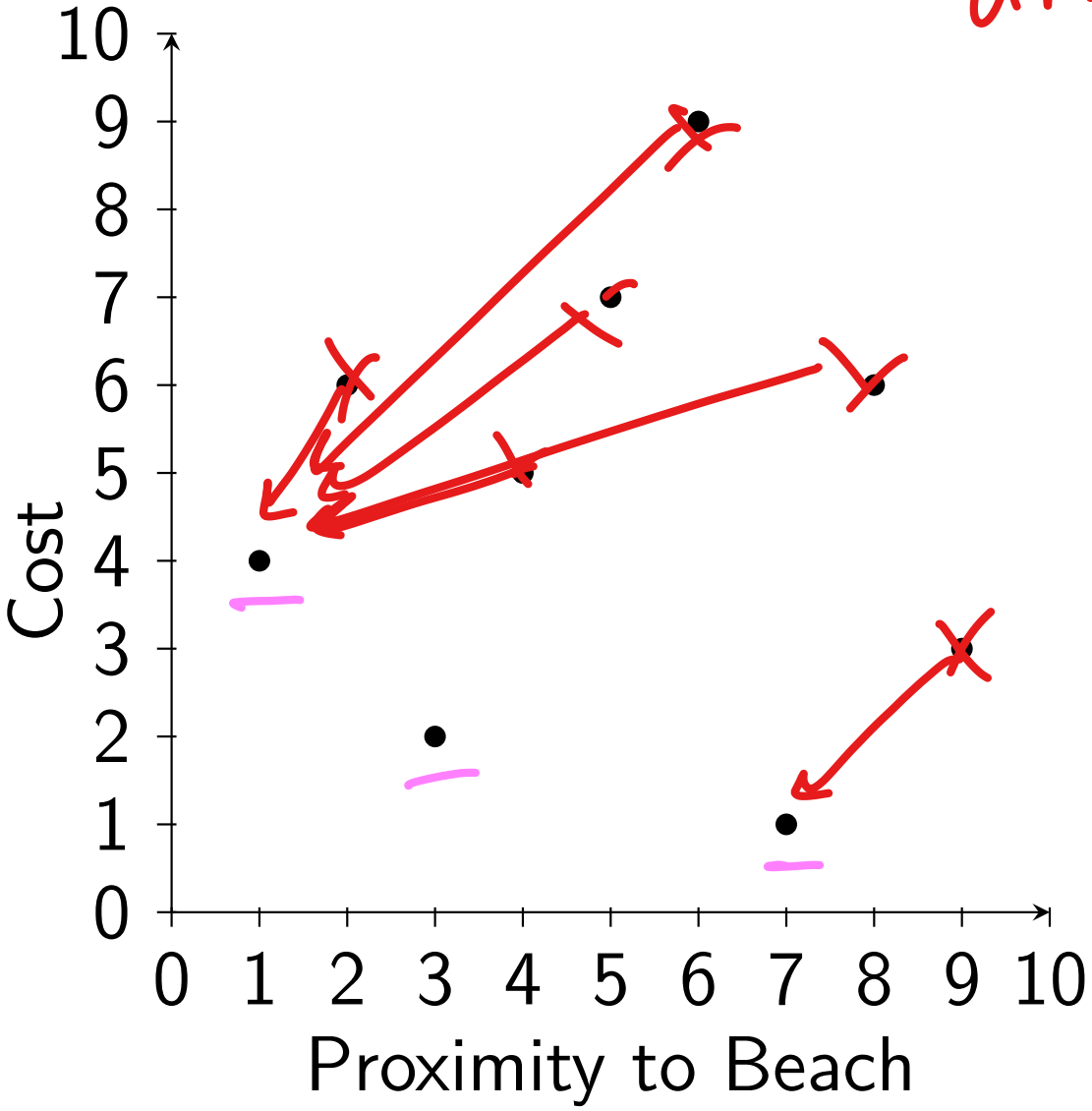
- Example: What is $13 \cdot 11$?

X_H	X_L	Y_H	Y_L
11	01	10	11
3	1	2	3

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.

Minima Set Example



arrow = one reason
why elim

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

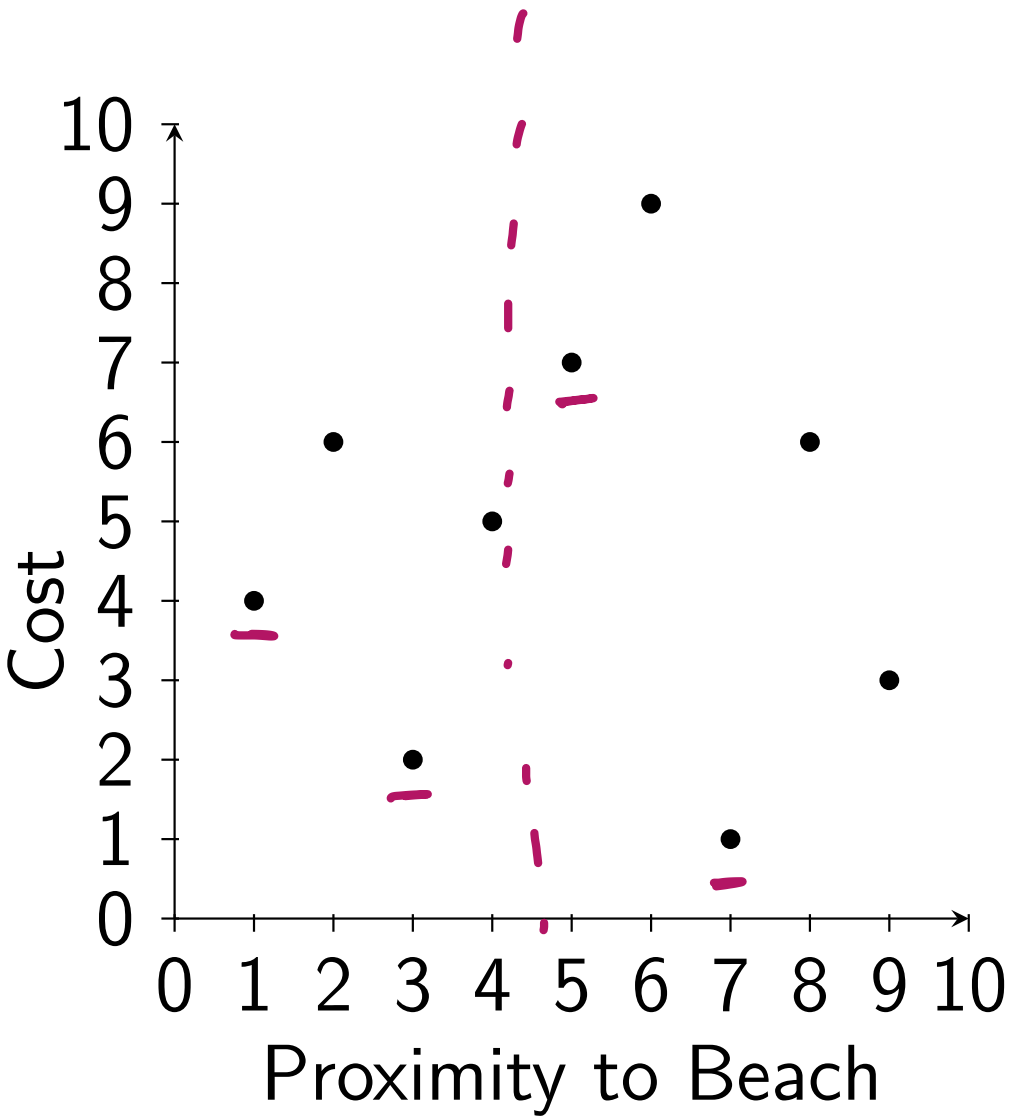
- This is $O(n^2)$.

Beginning Divide and Conquer

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

- return $M_1 \cup M_2$?

Example revisited



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

Finding a correct recombine

```
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)
```

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

Improved Recombine

$\Theta(n)$

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

Culprit

$M_1 \leftarrow$ `MinimaSet`(L)

$M_2 \leftarrow$ `MinimaSet`(G)

a = min-cost in M_1

~~**for** each $a \in M_1$ **do**~~

  for each $b \in M_2$ **do**

    if a better than b **then**

      remove b from M_2

$\Theta(n^2)$

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