COP3503 3/29/2022

Divide + Conquer (Recursion)

(1) Integer Multiplication

(2) Tromino Tiling

(3) Skyline Problem

Reg grade school

$$
\begin{array}{lr}
n & 236 \\
n & \times 417 \\
\hline
 & 1652 \\
 & 236 \\
 & +\,944 \\
\hline
 & 98412
\end{array}
$$

$O(n)$
$O(n)$
$\vdots$
$O(n)$

n times $\Rightarrow O(n^2)$

+

adding upto n digits no more than 2n times $\Rightarrow O(n^2)$

$O(n^2)$

Can we do better?

Pretend n-digit numbers are such that $n = 2^k$ for some pos. int k.

$$X = 2347 = 23 \times 10^2 + 47$$
$$Y = 8106 = 81 \times 10^2 + 06$$

$$(23 \times 10^2 + 47)(81 \times 10^2 + 06) = (23 \times 81)10^4 + (23 \times 06)10^2 + (47 \times 81)10^2 + 47 \times 06$$

When we foil out

$\Rightarrow$ 4 terms

$\Rightarrow$ each is a 2 digit mult.

$$I = I_h \times 10^{n/2} + I_L$$

$$J = J_h \times 10^{n/2} + J_L$$

$$I \times J = (I_h J_h) 10^n + \boxed{(I_h J_L + I_L J_H)} 10^{n/2} + I_L J_L$$

Let $T(n)$ = run time of this algorithm

$$T(n) = 4T(\tfrac{n}{2}) + O(n)$$

$A = 4,\ B = 2,\ k = 1$

$$B^k < A \Longrightarrow O(n^{\log_2 4}) \rightarrow O(n^2)$$

$$P_3 = (I_h + I_L)(J_h + J_L) = I_h J_h + I_L J_h + I_h J_L + I_L J_L$$

$$P_1 = I_h J_h, \quad P_2 = I_L J_L$$

$$I_h J_L + I_L J_H = P_3 - P_1 - P_2$$

Same recursive breakdown, but only calculate these 3 recursive calls

New Rec Rel $\quad T(n) = 3T(\tfrac{n}{2}) + O(n)$

$$\Longrightarrow O(n^{\log_2 3})$$

Karatsuba's Alg.

Smooth Int

82	43	99	6	48

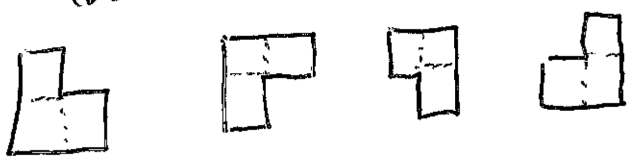
+5 +10 +1 +4

~~87~~ ~~53~~ ~~100~~ ~~10~~ 8

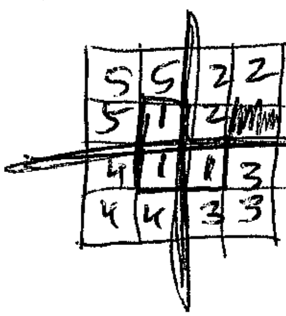
8 7 3 0 0

= 873008

Tromino Tiling



Input $2^n \times 2^n$ grid with a hole missing



goal to fill rest of squares with non-overlapping trominos

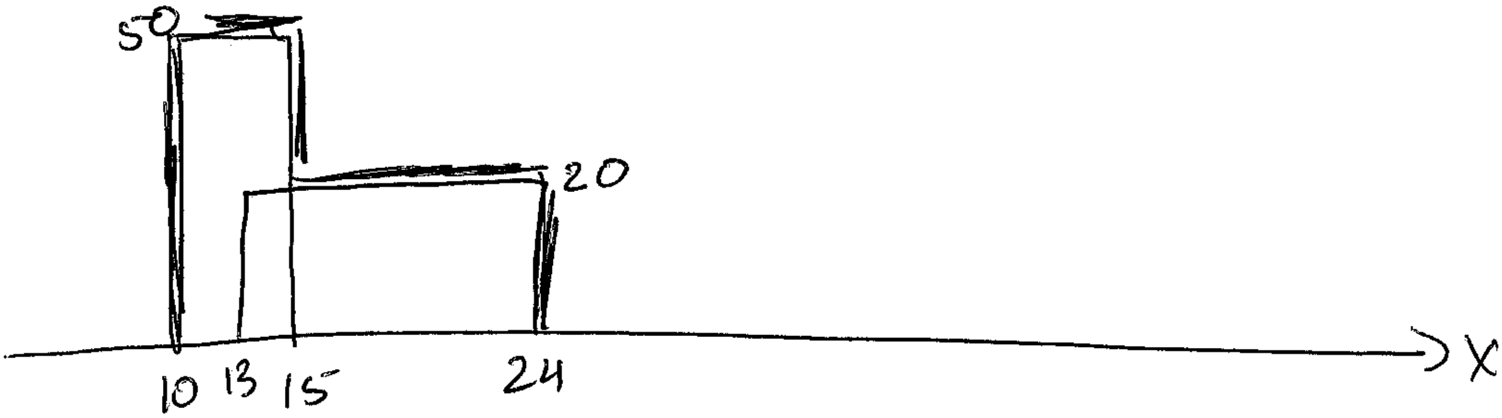
Idea: (1) Split into 4 $2^{n-1} \times 2^{n-1}$ grids

(2) One of the 4 quadrants is a valid recursive call (has a hole)

(3) Place one tromino in the middle in the 3 quadrants w/o a hole

(4) Now we can recursively tile each $2^{n-1} \times 2^{n-1}$ grid because they each have a hole!

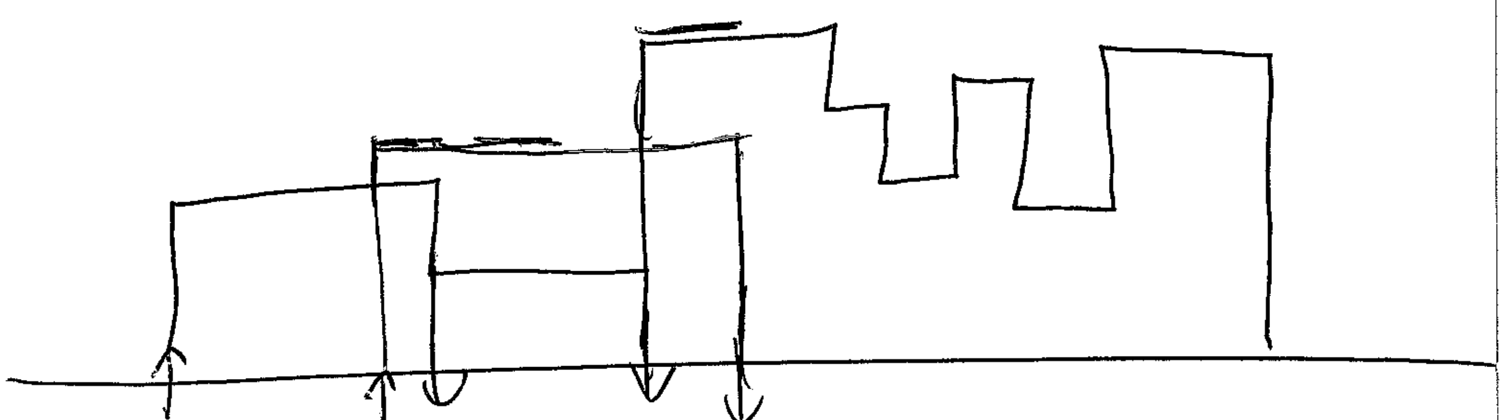
Skyline Problem



n buildings B_i has $xLow_i$, $xHigh_i$, height

(10, 15, 50)
(13, 24, 20)

Goal "compute" the skyline.
How to add a building to an existing skyline



Sweep L to R ends

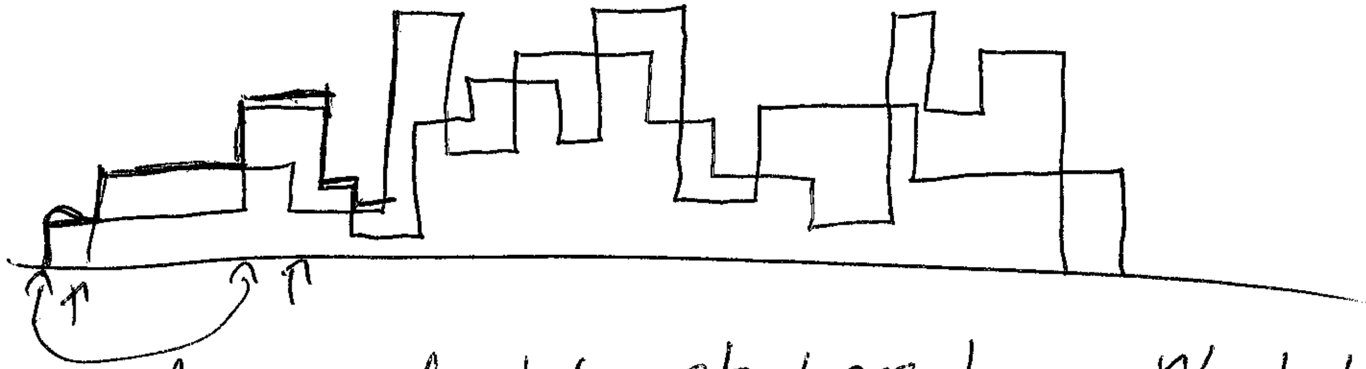
Run time is $O(n)$ if the existing skyline has n buildings.

```
for (i = 0 ; i<n; i++)
    merge (current, building[i])
```

$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = O(n^2)$

Can we do better?

What about merging 2 skylines?



Merge if both skylines have $n/2$ buildings

$\Longrightarrow O(n)$

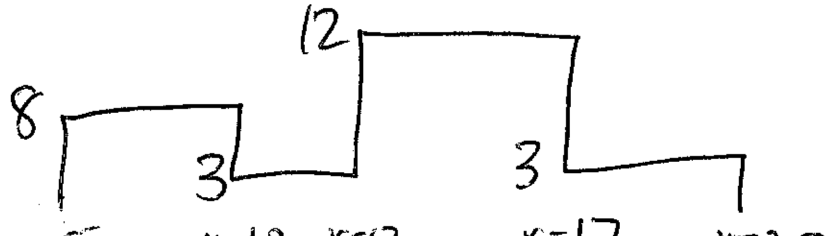
Idea - like Merge Sort

$S_1 = \text{Skyline}(0, n/2-1)$

$S_2 = \text{Skyline}(n/2, n-1)$

return $\text{Merge}(S_1, S_2)$

```
int[] merge(int[] sky1, int[] sky2) {
```



$$\begin{array}{ccccccccc}
(5, & 8, & 10, & 3, & 13, & 12, & 17, & 3, & 20) \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
x & h & x & h & x & h & x & h & x
\end{array}$$