CSC006P1M: Design and Analysis of Algorithms
Lecture 06 *(Divide-and-Conquer Paradigm)*

Sumit Kumar Pandey

August 29, 2022

Divide-and-Conquer Paradigm

In Divide-and-Conquer paradigm, we solve a problem recursively, applying three steps at each level of recursion.

1. **Divide** - The problem is divided into a number of smaller sub-problems that are smaller instances of the same problem.
2. **Conquer** - The sub-problems are conquered by solving them recursively. If the sub-problem sizes are small enough, just solve the sub-problems in straightforward manner.
3. **Combine** - The solutions to the sub-problems are combined into the solutions for the original problem.

Decrease-and-Conquer Paradigm

In Decrease-and-Conquer paradigm, we solve a problem recursively, applying three steps at each level of recursion.

1. **Decrease** - The problem instance is reduced to smaller instance of the same problem.
2. **Conquer** - The problem is solved by solving smaller instance of the problem.
3. **Extend** - The solution of smaller solutions is extended to obtain solution to original problem.

This approach is also known as incremental approach.

Divide-and-Conquer vs Decrease-and-Conquer

1. If each problem is divided into **two or more** sub-problems, then the approach is called **divide-and-conquer**.
2. If each problem is divided into **one** sub-problem, then the approach is called **decrease-and-conquer**.

Decrease-and-Conquer Paradigm

Variations of Decrease-and-Conquer

1. Decrease by a constant - In each iteration of the algorithm, the size of an instance is reduced by the same constant.
2. Decrease by a constant factor - In each iteration of the algorithm, the size of an instance is reduced by the same constant.
3. Variable size decrease - The size-reduction pattern varies from one iteration of an algorithm to another.

Some Previous Algorithms

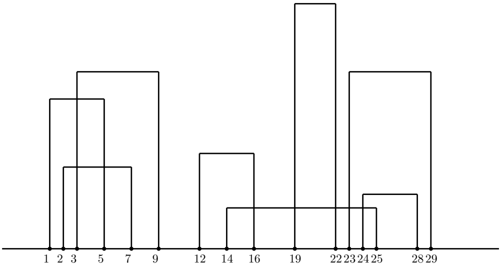
- Evaluating Polynomial (Version - 1) - Decrease-and-Conquer.
- Evaluating Polynomial (Version - 2) - Decrease-and-Conquer.
- Evaluating Polynomial (Horner's rule) - Decrease-and-Conquer.
- One-One Mapping - Decrease-and-Conquer.
- Gray Code (Version 1) - Decrease-and-Conquer.
- MAGNUS problem - Decrease-and-Conquer.
- Maximum Consecutive Sum - Decrease-and-Conquer.

An Example of a Divide-and-Conquer Algorithm

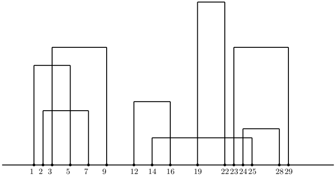
The Skyline Problem

Given the exact locations and shapes of several rectangular buildings in a city, draw the skyline (in two dimensions) of these buildings, eliminating hidden lines.

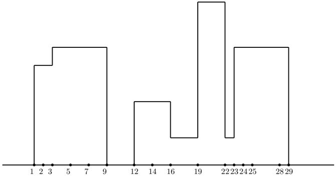
Example: $(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16)$, $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28)$.



The Skyline Problem



$(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16)$, $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28)$.

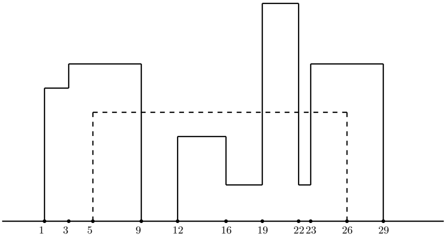


$(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, , \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$

Induction Hypothesis

We know how to solve the problem for $n - 1$ buildings.

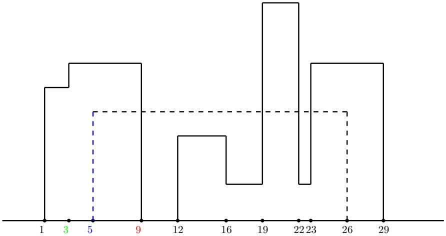
And now, we add the n^{th} building.
Let B_n be $(5, \mathbf{9}, 26)$.



The Skyline Problem

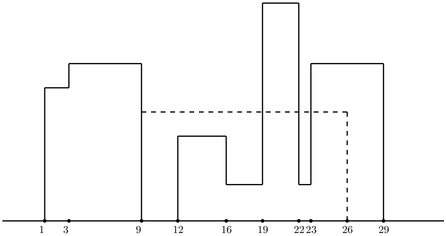
Previous Configuration:
$(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$
New Building: $B_n : (5, \mathbf{9}, 26)$.
Algorithm:

1. Scan the skyline from left to right and find where the left side of B_n fits. For example, in our case the co-ordinate 5 fits between 3 and 9. $((5, \mathbf{9}, 26);$
$(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0}))$



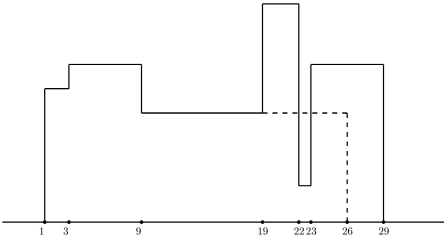
Algorithm:

2. Scan the horizontal line one after another, and adjust the height of B_n whenever the height of B_n is higher than the existing height.
    - The height of B_n at 5, which is **9**, is covered by the height of the existing skyline from 3 to 9, which is **13**. So, keep it same. $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$



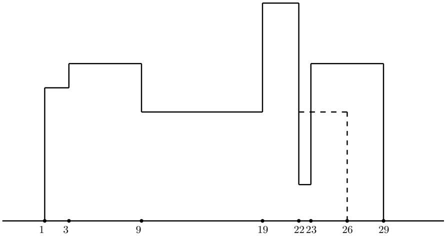
Algorithm:

3. Scan the horizontal line one after another, and adjust the height of B_n whenever the height of B_n is higher than the existing height.
   - The height of B_n from 9 till 19, which is **9**, is greater than the height of the existing skyline from 9 till 19, which are $\{\mathbf{0}, \mathbf{7}, \mathbf{3}\}$. So, change the configuration.
     $(1, \mathbf{11}, 3, \mathbf{13}, 9, \cancel{\mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$
     $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{9}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$



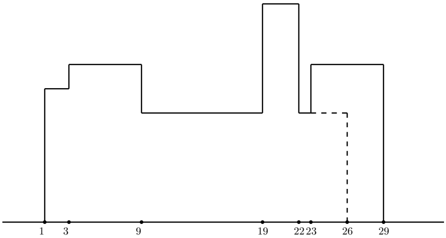
Algorithm:

4. Scan the horizontal line one after another, and adjust the height of B_n whenever the height of B_n is higher than the existing height.
   - The height of B_n from 19 till 22, which is **9**, is covered by the height of the existing skyline from 19 till 22, which is **18**. So, keep it same. $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{9}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29, \mathbf{0})$



Algorithm:

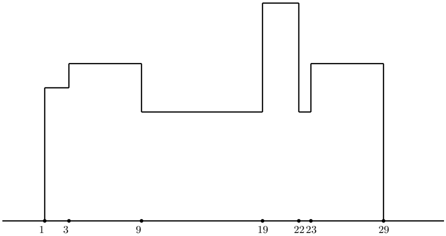
5. Scan the horizontal line one after another, and adjust the height of B_n whenever the height of B_n is higher than the existing height.
   - The height of B_n from 22 till 23, which is **9**, is greater than the height of the existing skyline from 22 till 23, which is **3**. So, change the configuration.
     $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{9}, 19, \mathbf{18}, 22, \cancel{\mathbf{3}}, 23, \mathbf{13}, 29, \mathbf{0})$
     $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{9}, 19, \mathbf{18}, 22, \mathbf{9}, 23, \mathbf{13}, 29, \mathbf{0})$



The Skyline Problem

Algorithm:

6. Scan the horizontal line one after another, and adjust the height of B_n whenever the height of B_n is higher than the existing height.
   - The height of B_n from 23 till 26, which is **9**, is covered by the height of the existing skyline from 23 till 26, which is **13**. So, keep it same. $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{9}, 19, \mathbf{18}, 22, \mathbf{9}, 23, \mathbf{13}, 29, \mathbf{0})$



How to Combine Pieces?

$(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16)$, $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28)$.

- $(1, \mathbf{11}, 5), (2, \mathbf{6}, 7) \rightarrow (1, \mathbf{11}, 5, \mathbf{6}, 7)$
- $(1, \mathbf{11}, 5, \mathbf{6}, 7), (3, \mathbf{13}, 9) \rightarrow (1, \mathbf{11}, 3, \mathbf{13}, 9)$
- $(1, \mathbf{11}, 3, \mathbf{13}, 9), (12, \mathbf{7}, 16) \rightarrow (1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16)$

$\vdots$

The Skyline Problem

Time Complexity:
$T(n) = T(n-1) + O(n)$
$T(n) = O(n^2)$

Why?

- $T(n) \leq T(n-1) + c \cdot n$
- $T(n-1) \leq T(n-2) + c \cdot (n-1)$

$$\vdots$$

- $T(2) \leq T(1) + c \cdot (2)$
- $T(1) = 0$

$T(n) \leq c(n + n - 1 + n - 2 + \cdots + 2) = c(n(n+1)/2 - 1)$.

Can we do better?

Use Divide-and-Conquer Approach.

The Skyline Problem

$(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16)$, $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28)$.

Divide them into two sub-problems:

1. $(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16)$.
2. $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28)$.

Conquer each sub-problems recursively.

1. $(1, \mathbf{11}, 5)$, $(2, \mathbf{6}, 7)$, $(3, \mathbf{13}, 9)$, $(12, \mathbf{7}, 16) \rightarrow$ $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16)$
2. $(14, \mathbf{3}, 25)$, $(19, \mathbf{18}, 22)$, $(23, \mathbf{13}, 29)$, $(24, \mathbf{4}, 28) \rightarrow$ $(14, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29)$

Combine the solutions of each sub-problems to get the final solution.

- $(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29)$.

Worst-Case Time Complexity:
$T_w(n) = 2T_w(n/2) + \Theta(n)$.

The Master Theorem

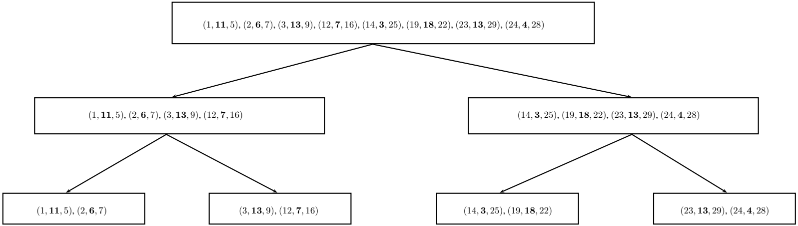
The solution of the recurrence relation $T(n) = aT(n/b) + cn^k$, where a and b are integer constants, $a \geq 1$, $b \geq 2$ and c and k are positive constants, is

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & \text{if } a > b^k \\ \Theta(n^k \log_b n) & \text{if } a = b^k \\ \Theta(n^k) & \text{if } a < b^k \end{cases}$$

$T_w(n) = \Theta(n \lg n)$.
$T(n) = O(n \lg n)$.

The Skyline Problem

Divide



Conquer

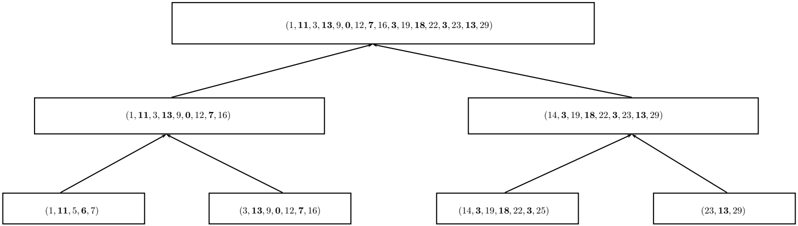
$(1, \mathbf{11}, 5, \mathbf{6}, 7)$

$(3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16)$

$(14, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 25)$

$(23, \mathbf{13}, 29)$

Combine


$(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29)$
$(1, \mathbf{11}, 3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16)$
$(14, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 23, \mathbf{13}, 29)$
$(1, \mathbf{11}, 5, \mathbf{6}, 7)$
$(3, \mathbf{13}, 9, \mathbf{0}, 12, \mathbf{7}, 16)$
$(14, \mathbf{3}, 19, \mathbf{18}, 22, \mathbf{3}, 25)$
$(23, \mathbf{13}, 29)$

Thank You