2019-Winter Sogang ACM

# Divide-and-Conquer

서강대학교 엄태경

# Divide and Conquer

**Divide and Conquer**

- Break up problems into parts
- Solve each part recursively
- Combine solutions of each part

**Examples**

- Merge sort
- Exponentiation by squaring ($a^n$)
- BOJ#1725 히스토그램

# Merge Sort (1)

**Pseudo-code**

```
MERGE-SORT(array):
    MERGE_SORT(left half of array)
    MERGE_SORT(right half of array)
    array = MERGE(left, right)

MERGE(left, right):
    result := empty array
    while left, right are not empty:
        add min(left[0], right[0]) to result
        and remove it from left/right
    if left or right is not empty:
        add it to result
    return result
```

**C++ implementation at https://github.com/thebarbershop/sogang-acmicpc-2019-winter**

# Merge Sort (2)

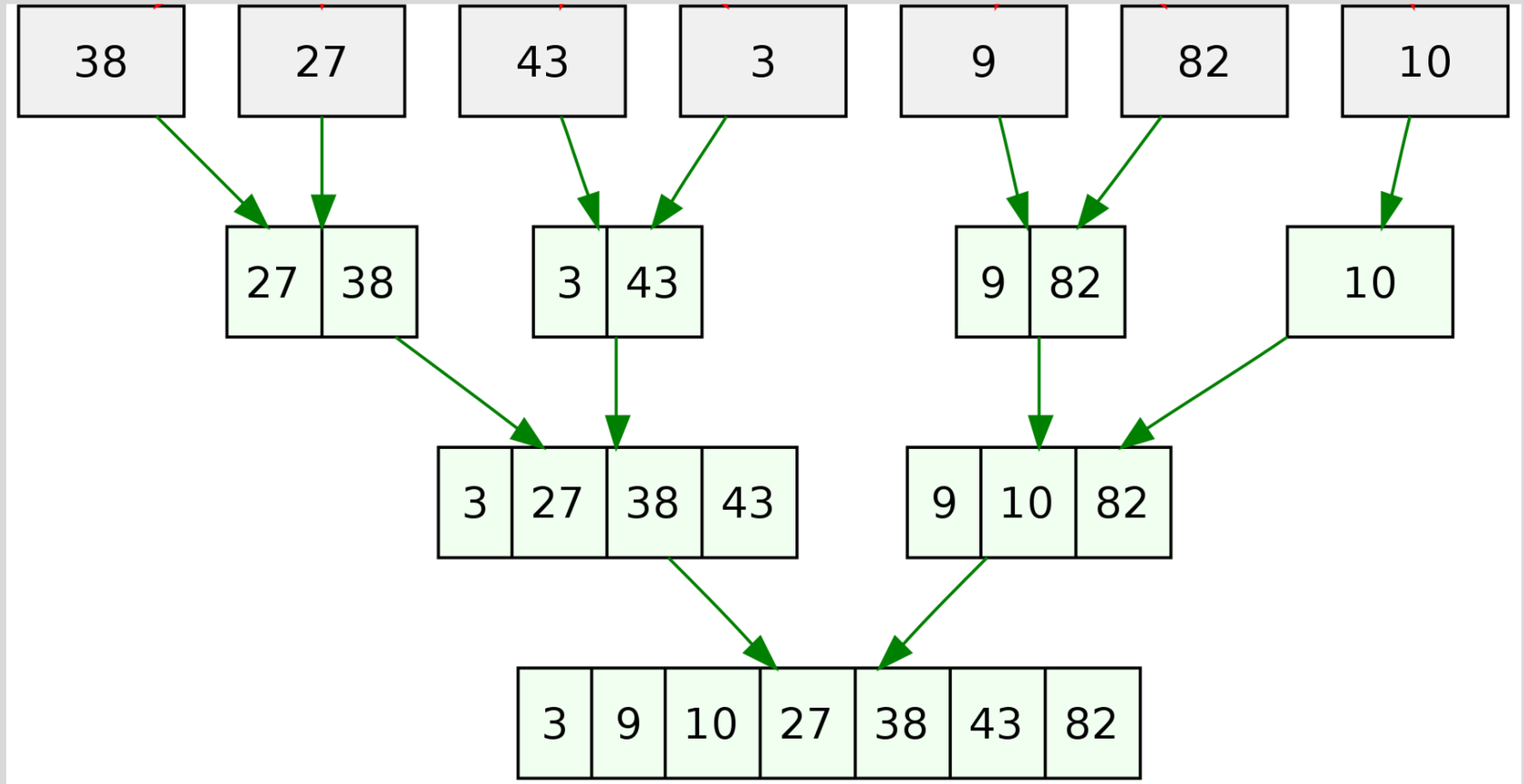
**Time Complexity**

```
MERGE-SORT(list m):
    MERGE_SORT(left)
    MERGE_SORT(right)
    MERGE(left, right)
```

- MERGE_SORT: only function call overhead
- MERGE: O(right-left)

# Merge Sort (3)

## Time Complexity




- Height of tree: O(log n)
- Cost of MERGE on each level: O(n)
- Time complexity: O(n log n)

# Exponentiation by squaring (1)



## Recursion

$$x^n = \begin{cases} x\left(x^2\right)^{\frac{n-1}{2}}, & \text{if } n \text{ is odd} \\ \left(x^2\right)^{\frac{n}{2}}, & \text{if } n \text{ is even.} \end{cases}$$

## Pseudo-code

```
POWER(x, n):
    if n == 0: return 1
    else if n == 1: return x
    else if n is odd:
        return x*POWER(x*x, (n-1)/2)
    else:
        return POWER(x*x, n/2)
```

C++ implementation at https://github.com/thebarbershop/sogang-acmicpc-2019-winter

# Exponentiation by squaring (2)



**Time Complexity**

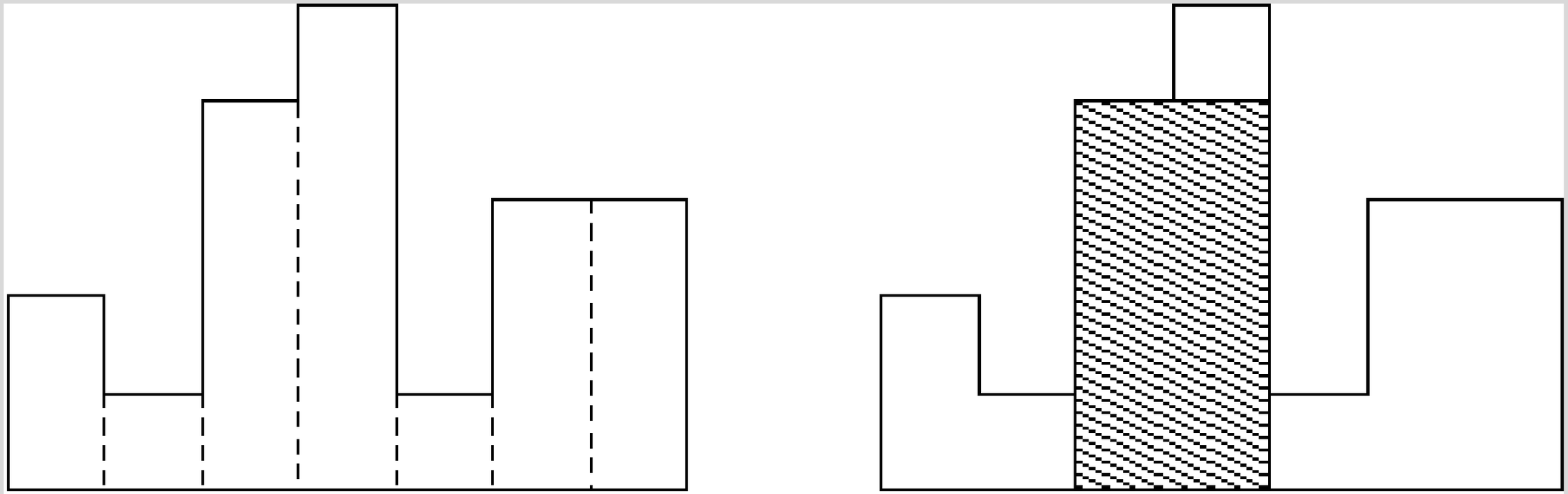
```
POWER(x, n):
    if n == 0: return 1
    else if n == 1: return x
    else if n is odd:
        return x*POWER(x*x, (n-1)/2)
    else:
        return POWER(x*x, n/2)
```

- Number of multiplication:
  (log n)+(# of 1-bit in n) = O(2 log n)
  = O(log n)

**C++ implementation at https://github.com/thebarbershop/sogang-acmicpc-2019-winter**

# Largest Rectangle in a Histogram

2019-Winter Sogang ACM



- Area of largest rectangle?
- Number of rectangles: 1 <= n <= 100,000
- Height of each rectangle: 0 <= $h_i$ <= 1,000,000,000
- Width of each rectangle: 1

**C++ implementation at https://github.com/thebarbershop/sogang-acmicpc-2019-winter**

# Practice



## BOJ

- 10827 a^b
- 10830 행렬 제곱
- 1780 종이의 개수
- 1725 히스토그램
- 2339 석판 자르기