INTRO TO DIVIDE + CONQUER

Learning Goals

- Describe divide + conquer structure [D&C2]
- Create recurrence relation for runtime of D&C [D&C1]
- Use tree formula to solve recurrence relation [D&C1]

Divide & Conquer Structure [D&C 2]

MergeSort

Input : Integer array A of length n
Output: Sorted array

```
// Base Case
if n == 1 then
    return A;
end
// Divide and Conquer
A_1 = MergeSort(A[1 : n/2]);
A_2 = MergeSort(A[n/2 + 1 : n]);
// Combine
p_1 = p_2 = 1;
for i=1 to n do
    if A_1[p_1] < A_2[p_2] then
        A[i] = A_1[p_1];
        p_1++;
    else
        A[i] = A_2[p_2];
        p_2++;
    end
end
```

Runtime Analysis [D&C 1]

① Identify variable that determines input size.

② $T(n)$ = runtime on input size n

MergeSort

Input : Integer array A of length n
Output: Sorted array

```
// Base Case
1  if n == 1 then
2  |  return A;
3  end
// Divide and Conquer
4  A_1 = MergeSort(A[1 : n/2]);
5  A_2 = MergeSort(A[n/2 + 1 : n]);
// Combine
6  p_1 = p_2 = 1;
7  for i=1 to n do
8  |  if A_1[p_1] < A_2[p_2] then
9  |  |  A[i] = A_1[p_1];
10 |  |  p_1++;
11 |  else
12 |  |  A[i] = A_2[p_2];
13 |  |  p_2++;
14 |  end
15 end
```

$$T(n) = \begin{cases} \text{\_\_\_\_\_} & \text{Base Case} \\ \text{\_\_\_\_\_} & \text{Recurrence} \end{cases}$$

Create recurrence relation for $T(n)$

Tree Formula [D&C 2]

$$T(n) = \begin{cases} O(1) & \text{if } n \leq 1 \\ 2T\left(\frac{n}{2}\right) + O(n) & \text{else } (n \geq 2) \end{cases}$$

If

$$T(n) = \begin{cases} O(1) & \text{if } n \leq c \\ aT\left(\frac{n}{b}\right) + O(n^d) & \text{else} \end{cases}$$

for c, a, b, d constants

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

What is the runtime of MergeSort?

A $O(n)$ B $O(n \log n)$ C $O(n^2)$ D $O(n^4)$