Class copy

Divide + Conquer Algorithms:

n = input of original

Step 1: Break my problem up into a subproblems
↳ a number!

Step 2: Use the recursion fxn on those sub-problems

Step 3: Put the answer together

f(n) we see in $T(n) = \sum T(k) + f(n)$
↑ subproblems

In-Class Exercise 05

CSCI 432

13 September 2023

Name(s):

If not handing in as one group, who did you work with today?

To solve this problem, I use a sub-problems of size $\frac{n}{b}$, and it takes f(n)-time to divide up into these sub-problems + bring them back together.

Master's Theorem

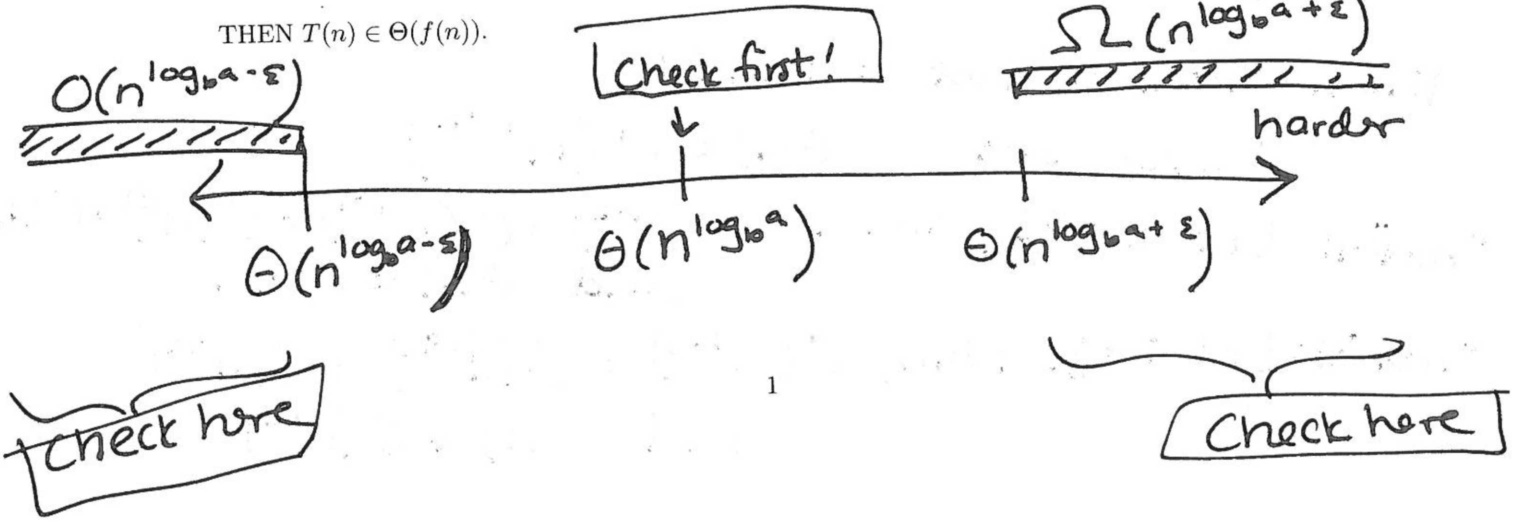
Master's theorem allows us to quickly solve recurrence relations of the form: a, b are constants ($\in \mathbb{Z}_+$)

$$\boxed{T(n) = aT(n/b) + f(n),}$$

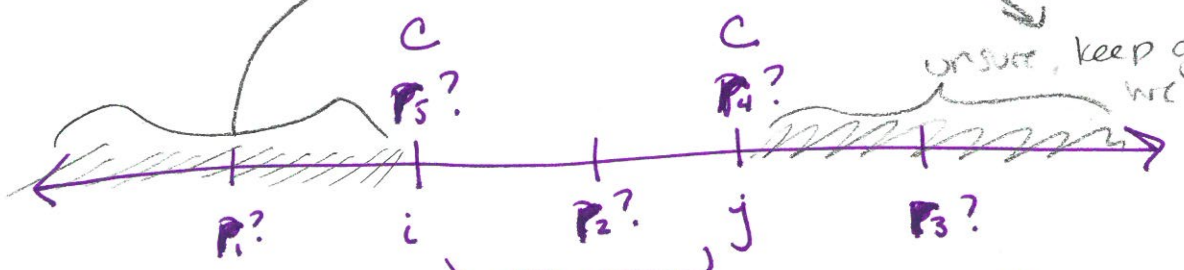
where $a, b \in \mathbb{N}$ such that $a \geq 1$ and $b > 0$ and $f(n)$ is asymptotically positive. Then, we can determine the closed-form of $T(n)$ as follows:

1. IF there exists $\varepsilon \in \mathbb{R}_+$ such that $f(n) \in O(n^{\log_b a - \varepsilon})$, THEN $T(n) \in \Theta(n^{\log_b a})$.
2. IF ~~there exists $\varepsilon \in \mathbb{R}_+$ such that~~ $f(n) \in \Theta(n^{\log_b a})$, THEN $T(n) \in \Theta(n^{\log_b a} \log n)$.
3. IF
   (a) there exists $\varepsilon \in \mathbb{R}_+$ such that $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ and ← same as b4
   (b) there exists $c \in (0, 1)$ and $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, the following holds: $af(n/b) \leq cf(n)$, ← regularity condition

   THEN $T(n) \in \Theta(f(n))$.



15 Sept 2023



unsure, keep going until we get a pivot in $\{i, i+1, \ldots, j-1, j\}$

Never compare i + j

2 ways they get compared out of $j-i+1$ ways we find out if they get compared.

```
QuickSort(A[1..n]):
1: if (n > 1)
2:     Choose a random pivot element A[p]
3:     r ← Partition(A, p)
4:     QuickSort(A[1..r−1])    ⟨⟨Recurse!⟩⟩
5:     QuickSort(A[r+1..n])    ⟨⟨Recurse!⟩⟩
6: return
```

```
Partition(A[1..n], p):
1: swap A[p] ↔ A[n]
2: ℓ ← 0                      ⟨⟨#items < pivot⟩⟩
3: for i ← 1 to n−1
4:     if A[i] < A[n]
5:         ℓ ← ℓ + 1
6:         swap A[ℓ] ↔ A[i]
7: swap A[n] ↔ A[ℓ+1]
8: return ℓ + 1
```

Figure 1.8. Quicksort

QuickSort $T(n)$ = RT of QS on $|A| = n$

Runtime: Line
1 $\Theta(1)$
2 $\Theta(1)$
3 $\Theta(n)$
4 $T(r-1)$
5 $T(n-r)$
6 $\Theta(1)$

$$T(n) = T(r-1) + T(n-r) + \Theta(n)$$

Worst-case is:

$$T(n) = \max_{1 \le i \le n} \{T(i-1) + T(n-i)\} + \Theta(n)$$

↳ Runtime?

Line 1, 2 } $\Theta(1)$

3, 4, 5, 6] For loop × (n−1) each line $\Theta(1)$ ⇓ $\Theta(n)$ total for these lines

7, 8 } $\Theta(1)$

$$\Theta(n) + 2\Theta(1) = \boxed{\Theta(n)}$$

1