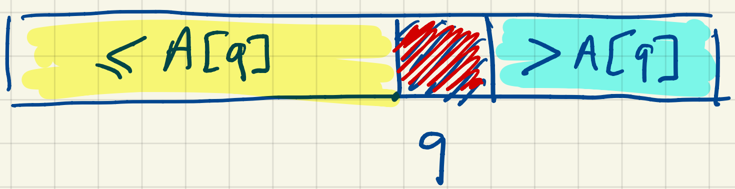
Quick Sort

Divide & Conquer that has a $\Theta(n^2)$ worst case running time but $\Theta(n \log n)$ expected (average) running time.

Divide: Partition $A[p \dots r]$ into $A[p \dots q-1]$ and $A[q+1 \dots r]$



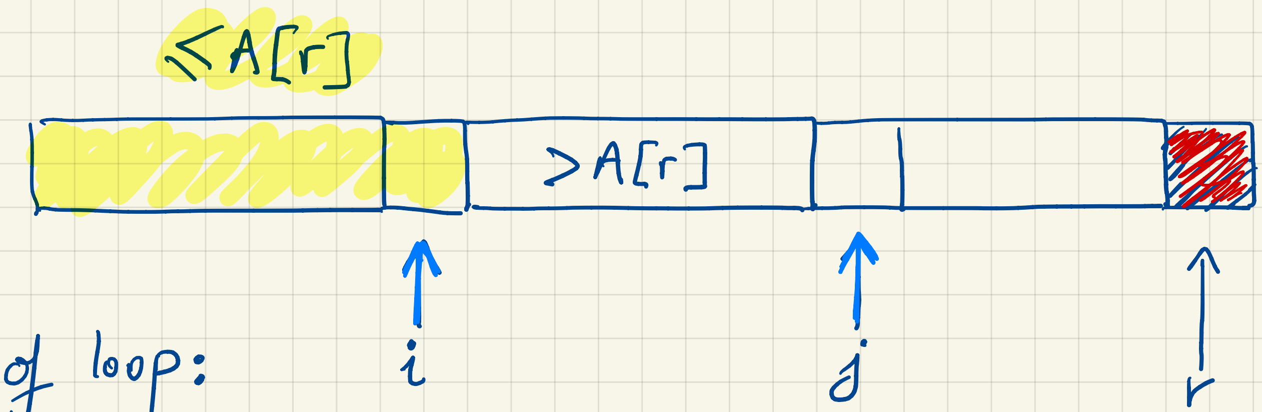
$A[q]$ is **Pivot**

Conquer: Recursively sort sub-arrays using Quicksort

Combine: No work is needed, already sorted in place

```
Quicksort (A, p, r)
    if p < r
        then q ← Partition (A, p, r)
             Quicksort (A, p, q-1)
             Quicksort (A, q+1, r)

Partition (A, p, r)
    x ← A[r]    ▷ the pivot is always last one
    i ← p-1     ▷ far left
    for j ← p to r-1
        do if A[j] ≤ x    (too small, move left)
            then i ← i+1
                 swap A[i] ↔ A[j]
    swap A[i+1] ↔ A[r]
    return i+1
```



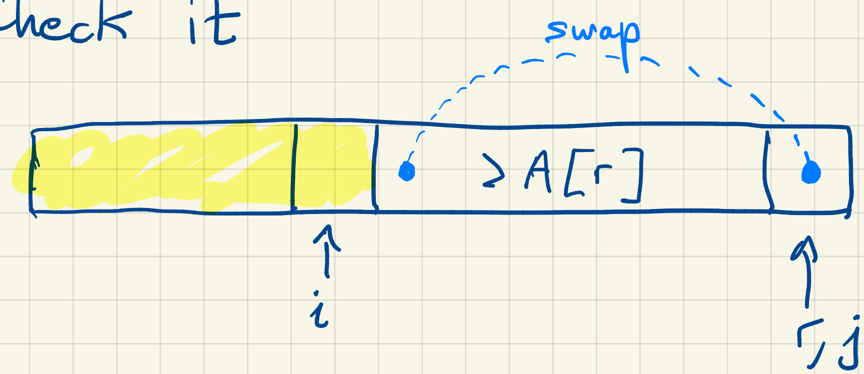
At beginning of loop:

1. $A[p \dots i] \le$ Pivot ✓
2. $A[i+1 \dots j-1] >$ Pivot ✓
3. $A[r] =$ Pivot ✓

<u>Init</u>: $A[p \dots i] = A[i+1 \dots j-1] = A[p \dots p-1]$ empty.

<u>Maintenance</u>: Check it

<u>Termination</u>:



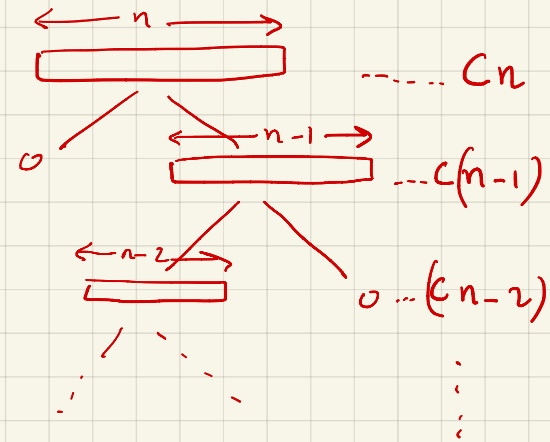
$j = r$

Some analysis:

worst case: $T(n) = T(n-1) + T(0) + \Theta(n)$

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

$$= \Theta(n^2)$$



$$T(n) = cn + T(n-1) = cn + c(n-1) + T(n-2) + \cdots$$

$$= c[n + (n-1) + (n-2) + \cdots + 1] = c\,\frac{n(n+1)}{2}$$

$$T(n) = \Theta(n^2)$$

Best case: $T(n) = 2T(n/2) + \Theta(n)$

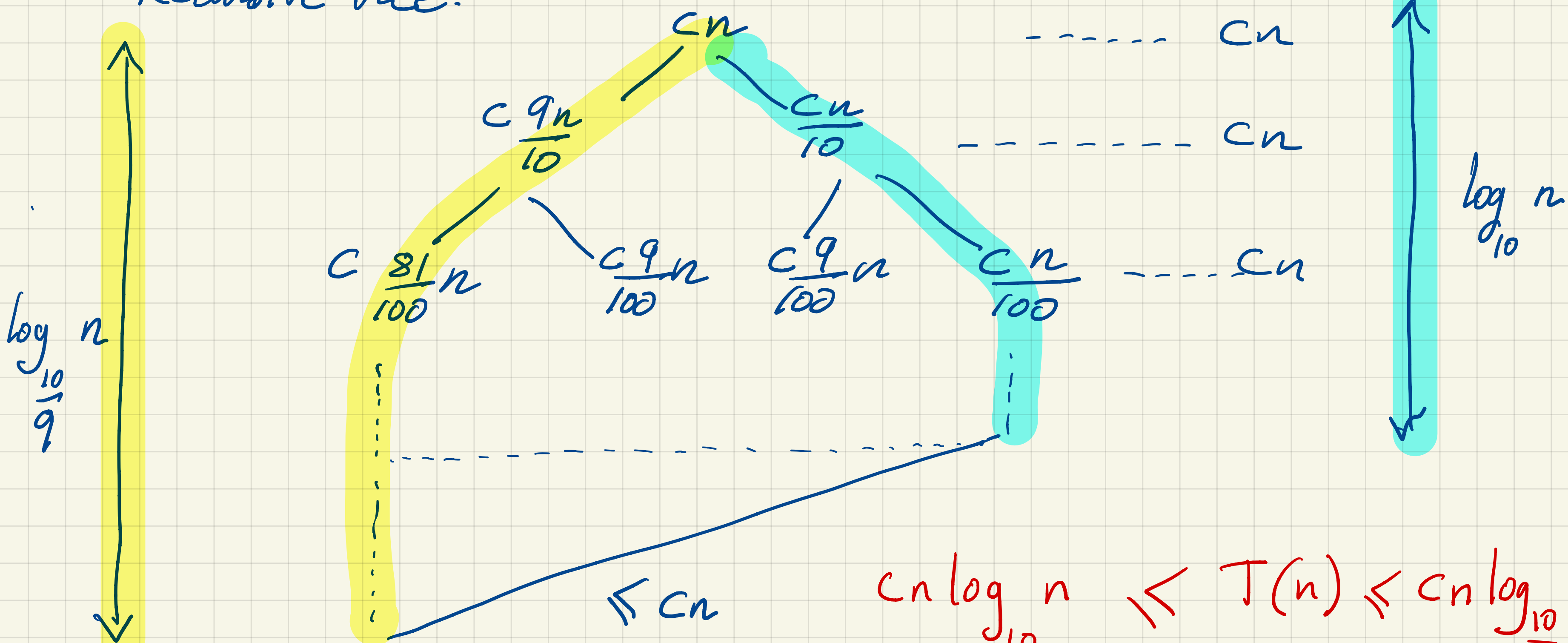
like mergesort $\Rightarrow T(n) = \Theta(n \log n)$

Quicksort average running time is much closer to the best case.

Intuition: Balanced partition. Assume a 9 to 1 split in worst case.

$$T(n) \le T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + \Theta(n) \Rightarrow T(n) = \Theta(n\log n)$$

Recursive tree:

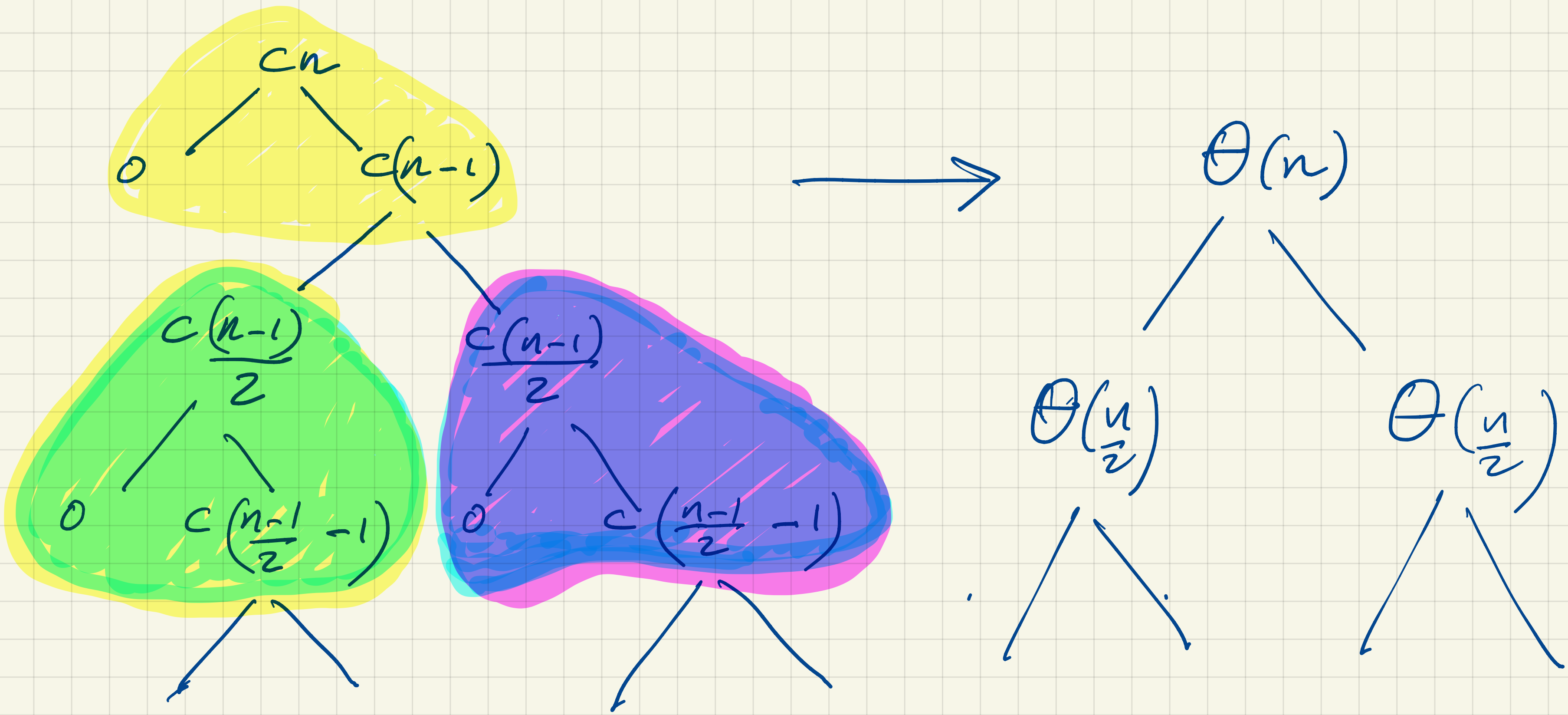


$$cn\log_{10} n \le T(n) \le cn\log_{\frac{10}{9}} n$$

$$T(n) = \Theta(n\log n)$$

What if split is not always balanced, e.g. a mix of good & bad splits.

Example: Assume it alternates between worst case & best case



Analysis: (worst case)

$$T(n) = \max_{0 \leq k \leq n-1} [T(k) + T(n-k-1)] + \Theta(n)$$

Assume $T(n) \leq dn^2$

$$T(n) \leq \max_{0 \leq k \leq n-1} \left[dk^2 + d(n-k-1)^2\right] + cn$$

$$= d \max_{0 \leq k \leq n-1} \left[k^2 + (n-k-1)^2\right] + cn$$

max occurs at $k=0$, or $k=n-1$

$$T(n) \leq d(n-1)^2 + cn \quad \left[\text{use } (a+b)^2 = a^2 + 2ab + b^2\right]$$

$$= dn^2 - d(2n-1) + cn \leq dn^2 \text{ if } d(2n-1) \geq cn$$

choose $d \geq \dfrac{cn}{2n-1}$

So $T(n) = O(n^2)$. [we can similarly show $\Omega(n^2)$]

RANDOMIZED Partition:

Rand Partition (A, p, r)

$i \leftarrow$ Random (p, r)

swap $A[r] \Leftrightarrow A[i]$

return Partition (A, p, r)

Change Quicksort to call Rand Partition.

$$T(n) = \frac{1}{n} \sum_{k=0}^{n-1} \left[T(k) + T(n-k-1) \right] + \Theta(n)$$

(Assume all partitions are equally likely)

$$T(n) = \frac{2}{n} \sum_{k=0}^{n-1} T(k) + \Theta(n) = \frac{2}{n} \sum_{k=2}^{n-1} T(k) + \underbrace{\frac{2}{n}T(0) + \frac{2}{n}T(1) + \Theta(n)}_{\Theta(n)}$$

Guess $T(n) \leq d n \log_2 n$, and absord $k=0$ and $k=1$ into the $\Theta(n)$

$$T(n) = \frac{2}{n} \sum_{k=2}^{n-1} T(k) + \Theta(n)$$

$$T(n) \le \frac{2}{n} \sum_{k=2}^{n-1} dk\log_2 k + cn$$

We can show $\sum_{k=2}^{n-1} dk\log_2 k \le \frac{1}{2} n^2 \log_2 n - \frac{1}{3} n^2$

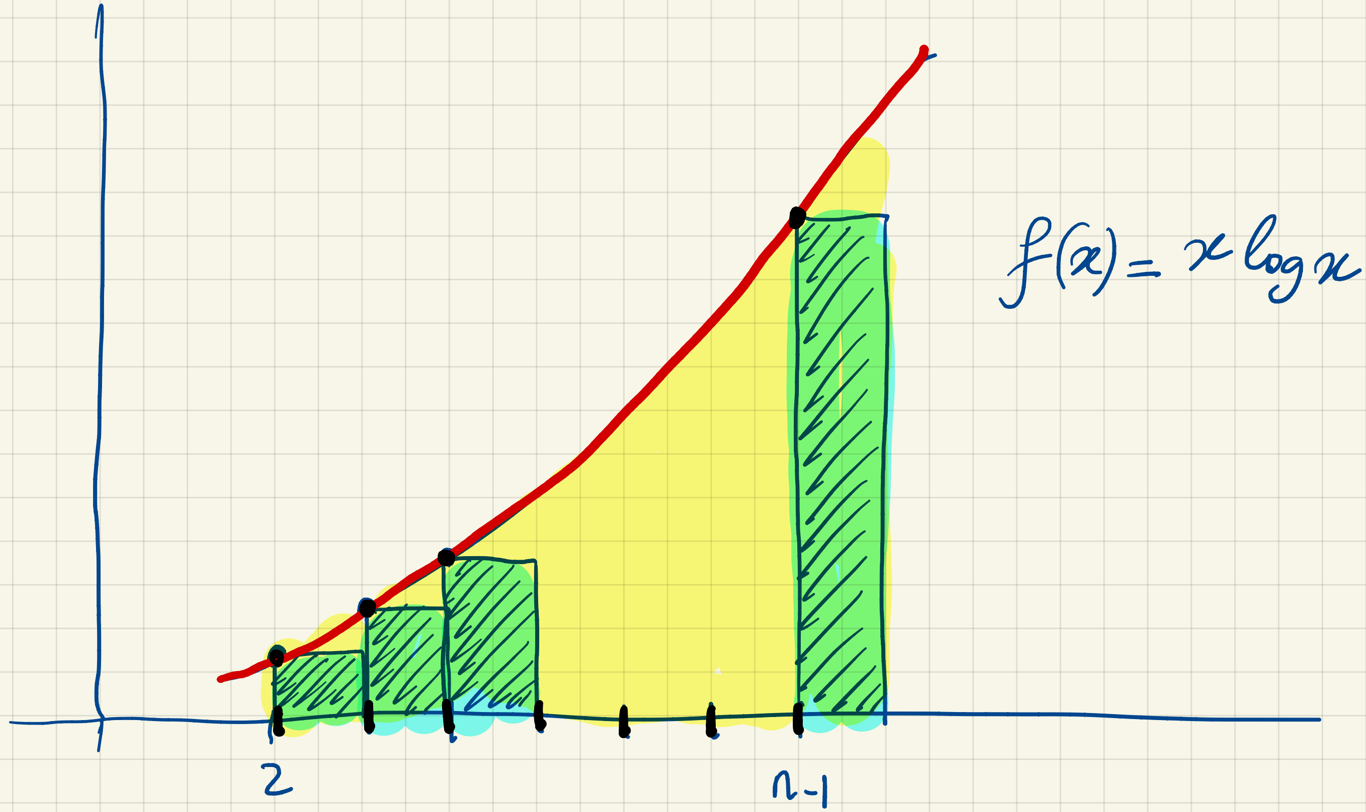
$$\boxed{\sum_{k=2}^{n-1} k\log_2 k \le \int_2^n x\log_2 x\,dx} = \frac{1}{\ln 2}\int_2^n x\ln x\,dx$$

$$= \frac{1}{\ln 2}\left[\frac{1}{2}x^2\ln x - \frac{x^2}{4}\right]_2^n$$

$$= \frac{1}{2}n^2\log_2 n - \frac{n^2}{4\ln 2} - \left(2 - \frac{1}{\ln 2}\right) \le \frac{1}{2}n^2\log_2 n - \frac{n^2}{3}$$

So $T(n) \le dn\log_2 n - \frac{2dn}{3} + cn$

$\le dn\log_2 n$ if $d \ge 1.5c$



Next time analysis using indicator random variables (Easier)