8]

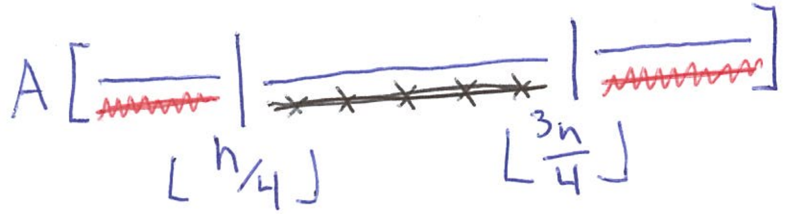
Cases: ① $i = k$ done

② $i > k$ search $L = A[0..i-1]$ for item k

③ $i < k$ search $R = A[i+1..n-1]$ for item $k-(i+1)$

$$T(n,k) = cn + \frac{1}{n}\Big[\overset{③}{\sum_{i=0}^{k-1} T\big((n-i-1), k-(i+1)\big)} + \overset{②}{\sum_{i=k+1}^{n-1} T(i,k)} + \overset{①}{0} \Big]$$

Lets be lazy: Define $T(n) = \max_{0 \le k < n} T(n,k)$



××× $\frac{1}{2}$ time pivot will be here
- subproblem size at most $\lfloor \frac{3n}{4} \rfloor$

〰 $\frac{1}{2}$ time pivot will be here
- subproblem size at most n

$$T(n) \le \begin{cases} cn + \frac{1}{2}\left[T(n) + T(\lfloor \frac{3n}{4} \rfloor)\right] & n \ge 2 \\ d & n = 1 \end{cases}$$

$$2T(n) \le 2cn + T(n) + T(\tfrac{3n}{4})$$

$$T(n) \le 2cn + T(\tfrac{3n}{4})$$

$$\le 2cn + 2c(\tfrac{3n}{4}) + 2c(\tfrac{9n}{16}) + \ldots + d$$

$$\le d + 2nc \underbrace{\sum_{i=0}^{\infty} (\tfrac{3}{4})^i}_{\Rightarrow 4} \in O(n)$$

* first partition is Ω (n

$\Rightarrow \Theta(n)$

Worst case: $\Theta(n^2)$

Best case: $\Theta(n)$

$\Rightarrow$ Average case: $\Theta(n)$

```
foo1(k)  // pre: 1 ≤ k ≤ 6
  for i = 1 to k
    printf("\uD83D\uDCA9")
```

$\boxed{10}$

average case: $\frac{1}{6}\sum_{i=1}^{6} i = \frac{7}{2}$

```
foo2
  k ← roll a 6-sided die
                  ↑ random
  foo1(k)
```

"expected" number of calls to print

$$\sum_{i=1..6} \underset{\uparrow\text{ probability}}{\left(\tfrac{1}{6}\right)} * \overset{\swarrow\text{ cost of case}}{i} = \frac{7}{2}$$

- uniform distribution

// pre: A[0..n-1] is a permutation of $\underbrace{1, 1, 1, \dots, 1}_{n-1 \text{ ones}}, n$

```
foo3(A, n)
 k = 0
 for i = 1 to A[k]
   printf("\uD83D\uDCA9")
```

$\boxed{1}$

#prints in best case?

#prints in worst case?

```
foo4(A, n)
  k ← random int in 0..n-1
  for i = 1 to A[k]
    print("\uD83D\uDCA9")
```

* k is random - sum over all cases

Let I_1 be $\overset{\text{fixed}}{\text{input}}$ $A[\overset{0}{1}\ \overset{1}{1}\ \overset{2}{4}\ \overset{3}{1}]$, $n=4$

$$T^{(exp)}(I_1) = \sum_{k=0..3} T(I_1, k) * \frac{1}{4}$$

$$= 1\left(\tfrac{1}{4}\right) + 1\left(\tfrac{1}{4}\right) + 4\left(\tfrac{1}{4}\right) + 1\left(\tfrac{1}{4}\right) = 1\tfrac{3}{4}$$

Let I_n be input $A = [\overset{0}{n}\ \overset{1}{1}\ \overset{2}{1} \cdots \overset{n-1}{1}]$

- input is fixed
- k is random

$\boxed{1}$

$$T^{(exp)}(I_n) = \sum_{k=0..n-1} T(I_n, k) * \left(\tfrac{1}{n}\right) \leftarrow \text{probability of each case}$$

- uniformly chosen

$$= 1 * \overset{\text{\# cases of 1}}{\frac{n-1}{n}} + \overset{\text{one n case}}{n}\left(\frac{1}{n}\right)$$

$$= \frac{2n-1}{n} < 2$$

Worst case Expected runtime

$$T^{(exp)}(n) = \max_{size(I)=n} T^{(exp)}(I)$$

- max of all inputs of a certain size

$$T^{(exp)}(n) = 1 * \frac{n-1}{n} + n\left(\frac{1}{n}\right)$$

$$= \frac{2n-1}{n} < 2$$

- each input of size n has $\begin{cases} n-1 & \text{ones} \\ 1 & n \end{cases}$

$T^{(exp)}(I) \sim$ for a particular instance

Worst Case : worst case for <u>any</u> problem instance of size n

*we use randomization to control the probability distribution

$\boxed{12}$ Randomize at beginning

- given a worst case => will get better - high probability
  - => could get worse - low probability

$\boxed{13}$ Random Pivot

- uniform distribution - 0, 1, ..., n-1 all equally likely
- $\frac{1}{2}$ time pivot is near the middle

$\boxed{14}$ BFPRT

- partitions array into groups of 5 with a few left over
- take median of each group of 5
  - ~ 2 elements smaller
  - ~ 2 elements bigger
- then take median of medians

lots of work to find pivot

- not practical to use

=> guarantee pivot is near the middle - or close enough

=> $\Theta(n)$ worst case

$$T(n) \leq \underbrace{T(\lceil \tfrac{n}{5} \rceil)}_{\text{median of medians}} + \underbrace{T(\tfrac{4}{5} n)}_{\text{recurse}} + \underbrace{O(n)}_{\text{partition}}$$

15 worst case

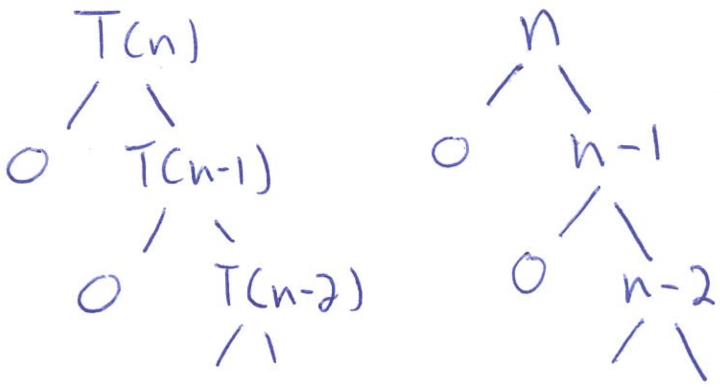
- array is sorted
- choose 0 as pivot

after partition: $A[\underbrace{\quad}_{\text{subproblem 1}}\overset{i}{\underset{\text{pivot}}{0}}\ \underbrace{1\ \ 2\ \dots\ n-1}_{\text{subproblem 2}}]$

$$T^{worst}(n) = T^{worst}(n-1) + \Theta(n)$$

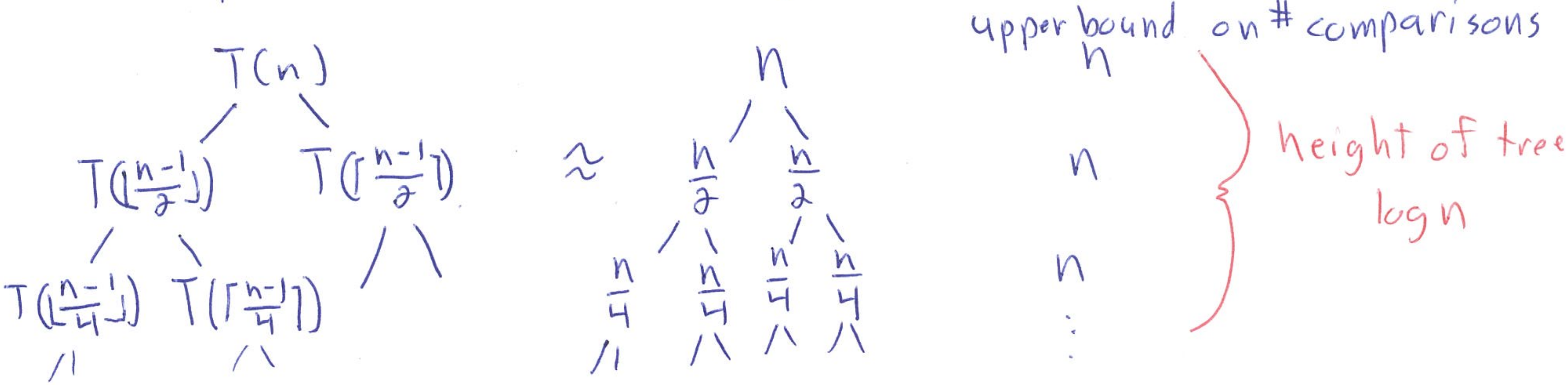
$$\Rightarrow \Theta(n^2)$$

Recursion Tree



Ideally want Balanced Partitioning

- subproblems of size: $\lfloor \frac{n-1}{2} \rfloor$ and $\lceil \frac{n-1}{2} \rceil$



Best case $\Rightarrow \Theta(n \log n)$

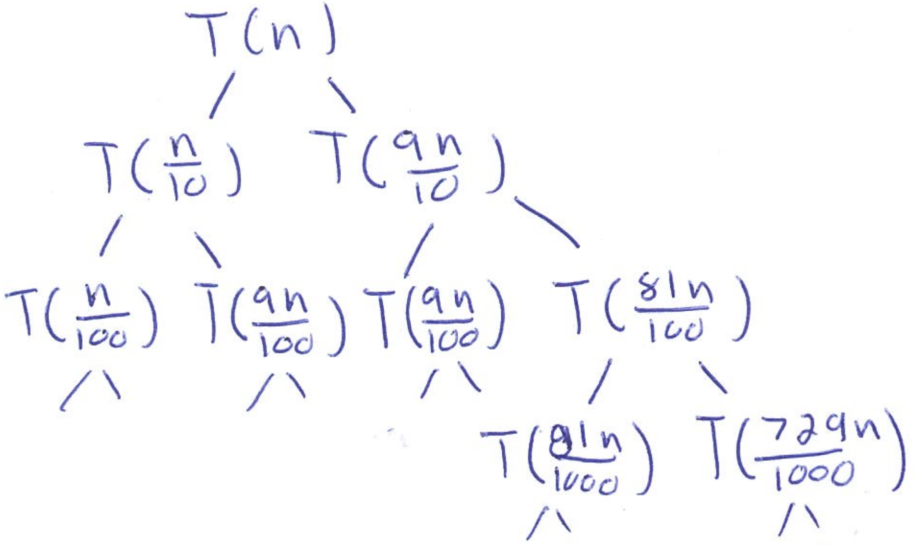
16) Average Case - usually not worst or best

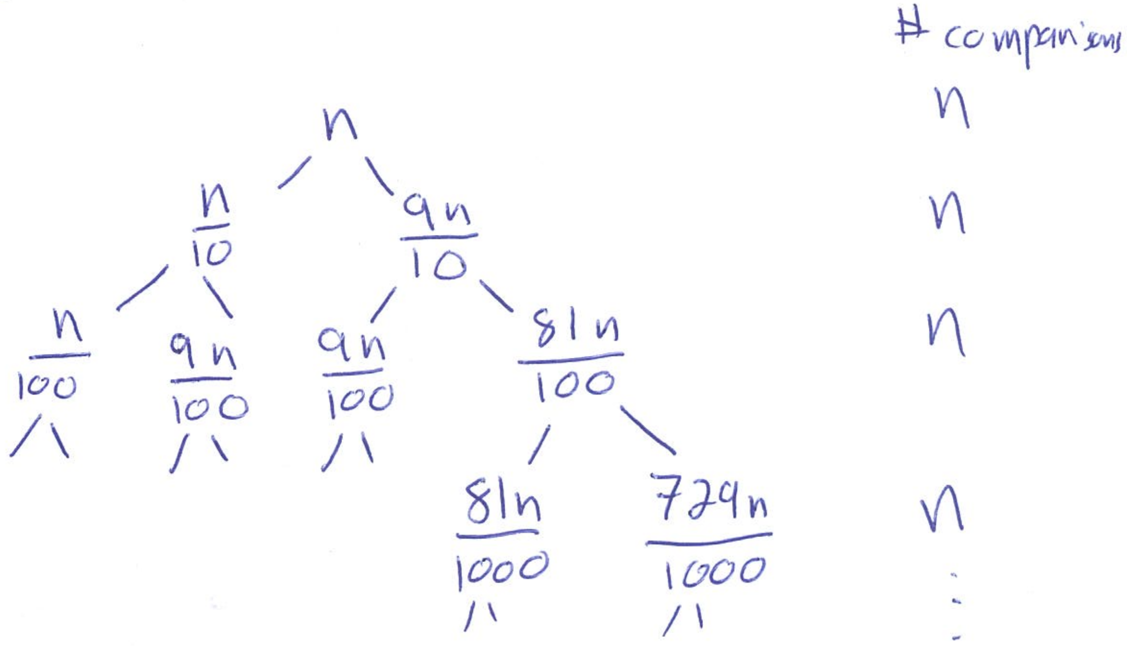
- maybe $\frac{1}{4}n$ and $\frac{3}{4}n$

Suppose partitioning always splits $\frac{1}{10}n$ and $\frac{9}{10}n$

$$\# \text{ comparisons } T(n) = \begin{cases} T(\frac{9n}{10}) + T(\frac{n}{10}) + n & \text{if } n > 1 \\ 0 & \text{if } n \le 1 \end{cases}$$

Recursion Tree





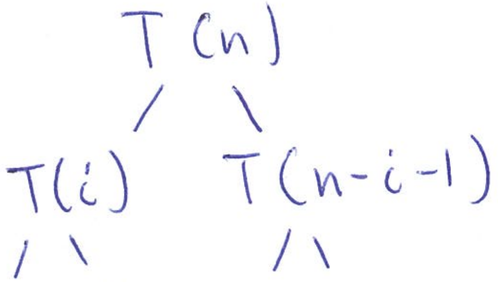
Deepest leaf?

Height of tree is minimal integer h such that:

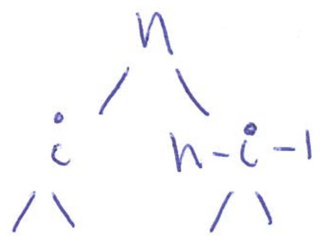
$$(\frac{9}{10})^h n \le 1 \Rightarrow h = \lceil \log (\frac{10}{9}) n \rceil \in O(\log n)$$

Total # comparisons $\in O(n \log n)$

Average Cast



=>



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

$\boxed{17}$ Let H(n) be average height of recursion tree

$$H(n) = \begin{cases} 1 + \frac{1}{n}\sum_{i=0}^{n-1} \max\left[H(i), H(n-i-1)\right] & n \geq 2 \\ 0 & n \leq 1 \end{cases}$$

$$H(n) \leq 1 + \frac{1}{2}H\left(\left\lfloor \frac{3n}{4} \right\rfloor\right) + \frac{1}{2}H(n)$$

$$2H(n) \leq 2 + H\left(\frac{3n}{4}\right) + H(n)$$

$$H(n) \leq 2 + H\left(\frac{3n}{4}\right) \leq 2 + 2 + H\left(\left(\frac{3}{4}\right)^2 n\right)$$

$$\overset{\circ\circ\circ}{\leq} 2 \cdot h \quad \text{where } h \text{ is minimal s.to } \left(\frac{3}{4}\right)^h \cdot n < 2$$

$\Rightarrow H(n) \in O(\log n)$

- Averagecase: $O(n\log n)$ } => Average Case $\Theta(n\log n)$
- Best Case: $\Theta(n\log n)$