Module 3

- see SortednessTester slides

Best-case : $\Theta(1)$

Worst-case : $\Theta(n)$

Average ?

Runtime is proportional to # of comparisons also # iterations of for loop

Let $T(\pi)$ be the # of comparisons

for some π, $T(\pi) = 1$ ≥ exactly 1 comparison

" $T(\pi) = 2$

$\vdots$

$T(\pi) = n-1$

$$T^{avg}(n) = \frac{1}{n!} \sum_{\pi \in \Pi_n} T(\pi)$$

$$T^{avg}(n) = \frac{1}{n!} \sum_{K=1}^{n-1} K \cdot (\text{\# of permutations with exactly } K \text{ comparisons})$$

---

perm with at least K comp

− # perm with at least K+1 comp

= # perm with exactly K comp

perm with at least k comp

k=1: at least 1 comparison => all $n!$

k=2: $A[0] < A[1]$ must be true

=> 0 and 1 occur in sorted order

eg $(4,3,2,0,1)$

$(0,3,1,4,2)$

$(3,0,2,4,1)$ ...

permutations: $\binom{n}{2}(n-2)!$

k=3: 0, 1, 2 occur in sorted order

$(4,0,3,1,2)$

$\binom{n}{3}(n-3)!$

k: 0, 1, 2, ..., k occur in sorted order

$$\binom{n}{k}(n-k)! = \frac{n!}{k!}$$

Taylor expansion $\sum_{k=0}^{\infty} \frac{1}{k!} = e \approx 2.8$

Let Π_K be $\frac{n!}{K!} \sim$ # of permutations with at least K comparisons

$$T^{avg}(n) = \frac{1}{n!}\left(\sum_{K=1}^{n-1} K \cdot \Pi_K - \sum_{K=1}^{n-1} K \cdot \Pi_{K+1}\right)$$

$$= \frac{1}{n!}\left(\begin{array}{l} 1 \cdot \Pi_1 + 2\Pi_2 + 3\Pi_3 + \dots + (n-1)\Pi_{n-1} \\ \quad - 1\Pi_2 - 2\Pi_3 - \dots - (n-2)\Pi_{n-1} \\ \quad\quad - (n-1)\Pi_n \end{array}\right)$$

$$= \frac{1}{n!}\left(\Pi_1 + \Pi_2 + \Pi_3 + \dots + \Pi_{n-1} \underbrace{-(n-1)\Pi_n}_{=0}\right)$$

$=0$ no cases with n comparisons

$$= \frac{1}{n!}\sum_{K=1}^{n-1} \Pi_K$$

$$= \frac{1}{n!}\sum_{K=1}^{n-1} \frac{n!}{K!} = \sum_{K=1}^{n-1} \frac{1}{K!} < 2.8$$

Average-case runtime of sortedness Tester is $O(1)$

Best-case: $\Theta(1)$

$\Rightarrow$ Average-case $\Theta(1)$

avg Case Demo

consider $n=5$, array A indices: 0..4

- indices 3 and 4 are last 2

  $\Rightarrow A[3]$ and $A[4]$

so if 3 comes before 4 in the sorting permutation $\Rightarrow$ good case

eg $\pi = (0, 1, 3, 2, 4)$

Bad case: not in order, 4 comes before 3

$\Rightarrow$ exactly half are good and half bad

$n!/2$ $n!/2$

$\overline{T}_{avg}$ is the sum over all permutations

$T(n)$ or $T(\pi)$ = only looking at one instance or one π so not avg.

avg Case Demo

Claim: $T^{avg}(n) \le 2\log n$

Pf by Induction

Base Case: $n \le 2$, no recursion $T^{avg}(n) = 0$

I.H.: Assume $n \ge 3$ and the claim holds for all $m < n$

$$T^{avg}(n) = 1 + \frac{1}{2}\underbrace{T^{avg}\left(\frac{n}{2}\right) + \frac{1}{2}T^{avg}(n-2)}_{\text{use I.H.}}$$

$$\le 1 + \cancel{\frac{1}{2}}\left(\cancel{2}\log\frac{n}{2}\right) + \cancel{\frac{1}{2}}\left(\cancel{2}\log(n-2)\right)$$

$$\le 1 + ((\log n) - 1) + \log n$$

$$= 2\log n \in O(\log n)$$

compared with worst-case $\Theta(n)$

$\log n$
$= \log \frac{n}{2} \cdot 2$
$= \log \frac{n}{2} + \log 2$
$= \log \frac{n}{2} + 1$
$\Rightarrow \log \frac{n}{2} = \log n - 1$

Average-Case Analysis for QuickSelect

Suppose you are given a sorting permutation π:

$$\begin{array}{l} \quad\; {\scriptstyle 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5} \\ \text{eg } \pi = (2, 0, 5, 1, 4, 3) \end{array}$$

Which item is chosen as the pivot?

- one in the last index of A A[5]

Where will this pivot end up at? (which index?)

2

How many other sorting permutations will place the pivot at the same index?

$(n-1)!$

$\Rightarrow$ there are n possible pivot locations
each one has $(n-1)!$ sorting permutations

$$T_{avg}(n) = \frac{1}{n!} \sum_{\pi \in \Pi_n} T(n)$$

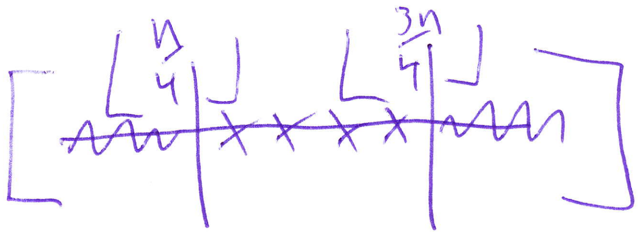
$$= \frac{1}{n!} \sum_{i=0}^{n-1} (\text{Runtime when pivot location is } i) * (\# \text{ permutations where pivot ends up at } i)$$

pivot locations $\rightarrow$; $\leftarrow (n-1)!$

$$= \frac{1}{n} \sum_{i=0}^{n-1} (cn + \max\{T(i), T(n-i-1)\}$$

$$= cn + \frac{1}{n} \sum_{i=0}^{n-1} \max\{T(i), T(n-i-1)\}$$

Instead of solving n cases, we'll group them into 2 groups



half of the instances will have a pivot-index $\sim\sim\sim$ and half of the instances it will be $\times\times\times$

For each group, choose the largest possible sub problem to recurse on:

$\sim\sim\sim \Rightarrow T(n)$ ← largest possible subproblem so finding O

$\times\times\times \Rightarrow T\left(\frac{3n}{4}\right)$

$$T(n) \leq \begin{cases} cn + \frac{1}{2}T(n) + \frac{1}{2}T\left(\frac{3n}{4}\right) & \text{if } n > 1 \\ d & \text{if } n \leq 1 \end{cases}$$

Pf

$$T(n) \leq cn + \frac{1}{2}T(n) + \frac{1}{2}T\left(\frac{3n}{4}\right)$$

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

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

$$T\left(\frac{3n}{4}\right) \leq 2c\left(\frac{3n}{4}\right) + T\left(\frac{9n}{16}\right)$$

$$T(n) \leq 2cn + 2c\left(\frac{3n}{4}\right) + T\left(\frac{9n}{16}\right)$$

$$\leq 2cn + 2c\left(\frac{3n}{4}\right) + 2c\left(\frac{9n}{16}\right) + T\left(\frac{27n}{64}\right)$$

$$\vdots$$

$$T(n) \leq \underbrace{d}_{\text{constant}} + 2nc \underbrace{\sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i}_{\Rightarrow 4} \in O(n)$$

Best case: $\Theta(n)$

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

Randomize QuickSelect

Claim: $T^{exp}(n) \in O(n)$, Show $T^{exp}(n) \leq 4cn$

Base case: $n=1$, $T(1) = c \leq 4c \cdot 1$

I.H. Assume $T(k) \leq 4ck$ for all $k < n$

$$T(n) \leq c \cdot n + \frac{1}{n}\sum_{i=0}^{n-1} \max\{\underbrace{T(i), \; T(n-i-1)}_{\text{apply I.H.}}\}$$

$$\leq cn + \frac{1}{n}\sum_{i=0}^{n-1}\max\{4ci, 4c(n-i-1)\}$$

$$\leq c \cdot n + \frac{4c}{n}\boxed{\sum_{i=0}^{n-1}\max\{i, n-i-1\}}$$

$$\leq cn + \frac{\cancel{4}c}{n}\cdot\frac{3n^2}{\cancel{4}} \in O(n)$$

$$= \sum_{i=0}^{\frac{n}{2}-1}\max\{i, n-i-1\} + \sum_{i=\frac{n}{2}}^{n-1}\max\{i, n-i-1\}$$

$$= \max\{0, n-1\} + \max\{1, n-2\} + \ldots + \max\{\tfrac{n}{2}-1, \tfrac{n}{2}\}$$

$$+ \max\{\tfrac{n}{2}, \tfrac{n}{2}-1\} + \ldots + \max\{n-1, 0\}$$

$$= n-1 + n-2 + \ldots + \frac{n}{2} + \frac{n}{2} + \ldots + n-1$$

pair

add pairs

$$= \left(n-1+\frac{n}{2}\right)\frac{n}{2} = \left(\frac{3n}{2}-1\right)\frac{n}{2} \leq \frac{3n^2}{4}$$

Claim: $T_B^{exp}(n) = T_A^{avg}(n)$

A: sloves Selection or Sorting

B: I: given instance
Randomly (and uniformly) permute $I \Rightarrow I'$
Call A on I'

- I is randomly & uniformly permuted
  $\Rightarrow$ with equal probability, I' is any instance of the problem. ~ $n!$ possible instances

$$T_B^{exp}(I) = \frac{1}{n!} \sum_{\substack{\text{all instances} \\ \text{of size } n}} (\text{runtime of } A) = T_A^{avg}(n)$$

one instance
But all instances have the same runtime

QuickSort

Worst-case: pivot falls at end of array
- subproblem of size $n-1$ – only 1 subproblem

Recursion Tree $\qquad T(n) = cn + T(n-1)$

$$
\begin{array}{ccc}
T(n) & & cn \\
\backslash & & \backslash \\
T(n-1) & \Rightarrow & c(n-1) \\
\backslash & & \backslash \\
T(n-2) & & c(n-2) \\
\vdots & & \vdots \\
T(1) & & 2
\end{array}
$$

$c(1+2+\dots+n) \in \Theta(n^2)$

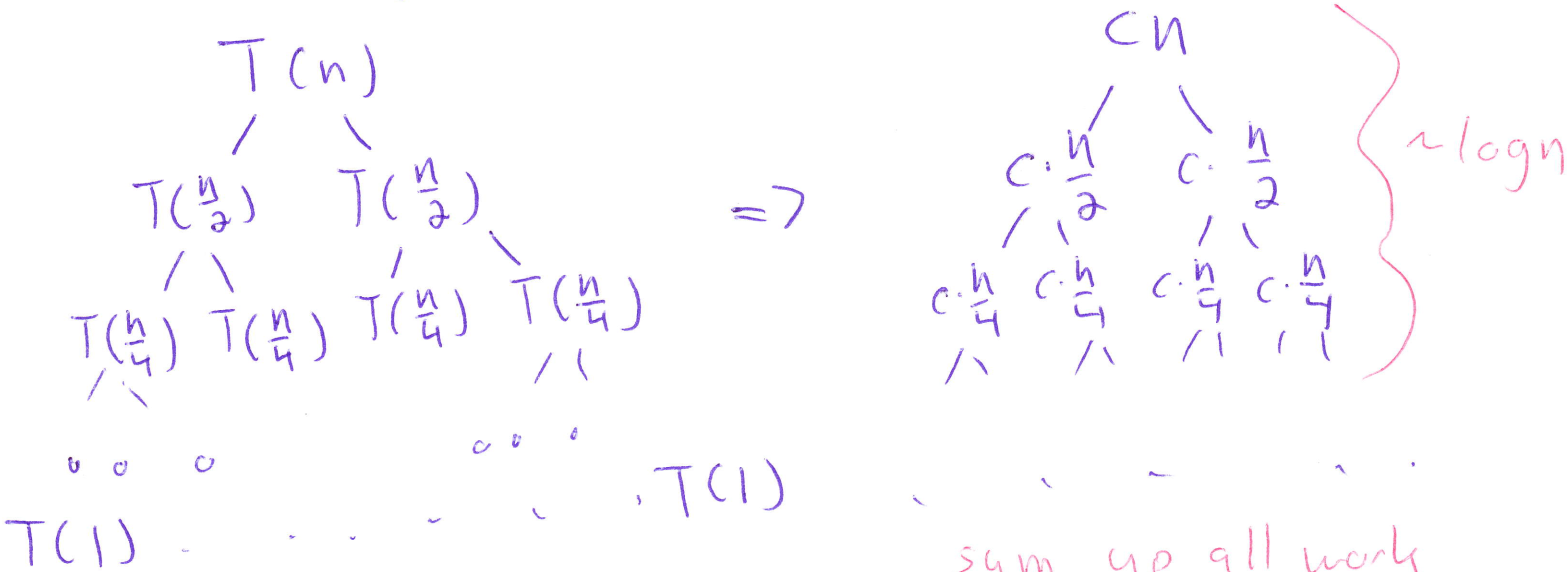
Best-case: want left and right branches of the recursion tree to be balanced.

Last time: Quicksort

- Ideally want pivot to fall in the middle

$\Rightarrow$ 2 subproblems of size $\frac{n}{2}$ or $\lfloor \frac{n-1}{2} \rfloor$ & $\lceil \frac{n-1}{2} \rceil$

Recursion Tree



sum up all work
- each level is $c \cdot n$
- How many levels?

$$T(n) = 2T\left(\frac{n}{2}\right) + cn$$

Height

"= 1" if power of 2

$$n \cdot \left(\frac{1}{2}\right)^h \leq 1$$

$$\log n + h \log \underbrace{\left(\frac{1}{2}\right)}_{=2^{-1}} \leq \log 1$$

$$\log n - h \cancel{\log 2} \leq 0$$

$$h \geq \lceil \log n \rceil \in \Theta(\log n)$$

careful if not power of 2
- ceiling, floor, add'l

height: $\Theta(\log n)$

Work per level: cn $\Rightarrow$ $c \cdot n \log n + n \cdot d$

n base cases: d $\in \Theta(n \log n)$

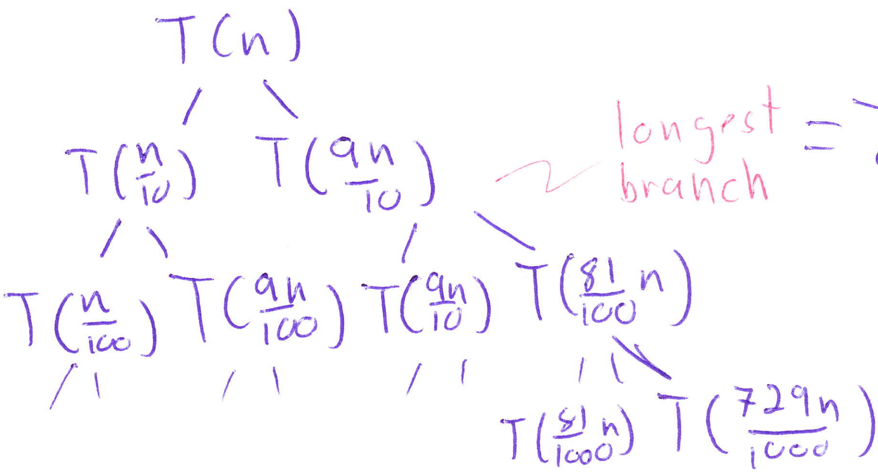
constant

Average Case: • not best, not worst

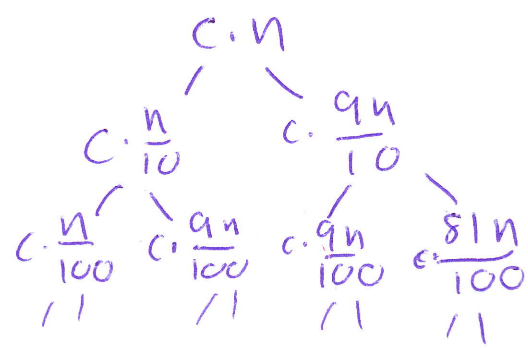
• maybe pivot falls between $\frac{1}{4}n$ and $\frac{3n}{4}$

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

$$\#\text{ comparisons } T(n) = \begin{cases} T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + cn \\ 0 \end{cases}$$



longest branch =>



• each level: $c \cdot n$

• How many levels?

• longest branch

Deepest Leaf?

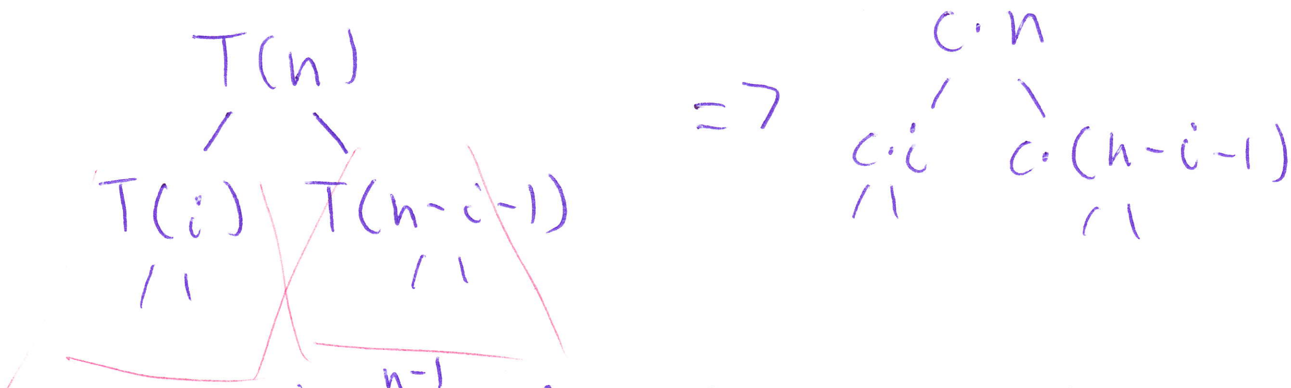
Height: $\left(\frac{9}{10}\right)^h \cdot n \leq 1 \Rightarrow h = \lceil \log_{(\frac{10}{9})} n \rceil$

$\in O(\log n)$

$\Rightarrow$ # levels: $O(\log n)$

upper bound work per level: $c \cdot n \Rightarrow O(n \log n)$

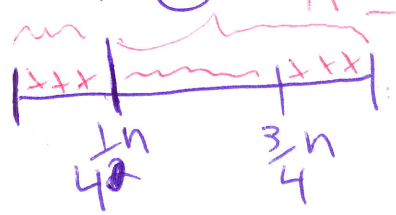
Average Case (Similarly Expected for Random pivot)



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

Let $H(n)$ be average (expected) height of the recursion tree

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



upper bounds

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

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

$$H(n) \leq 2 + H\left(\frac{3n}{4}\right) \leq 2 + 2 + H\left(\frac{9n}{16}\right)$$

$$\dots$$

$\leq 2 \cdot h$ where h is minimal s.t. $\left(\frac{3}{4}\right)^h \cdot n < 2$

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

• each level $O(cn)$ work $\Rightarrow O(n \log n)$

Best case : $\Theta(n \log n)$
Average : $O(n \log n)$ $\Big\} \Rightarrow$ Average case : $\Theta(n \log n)$

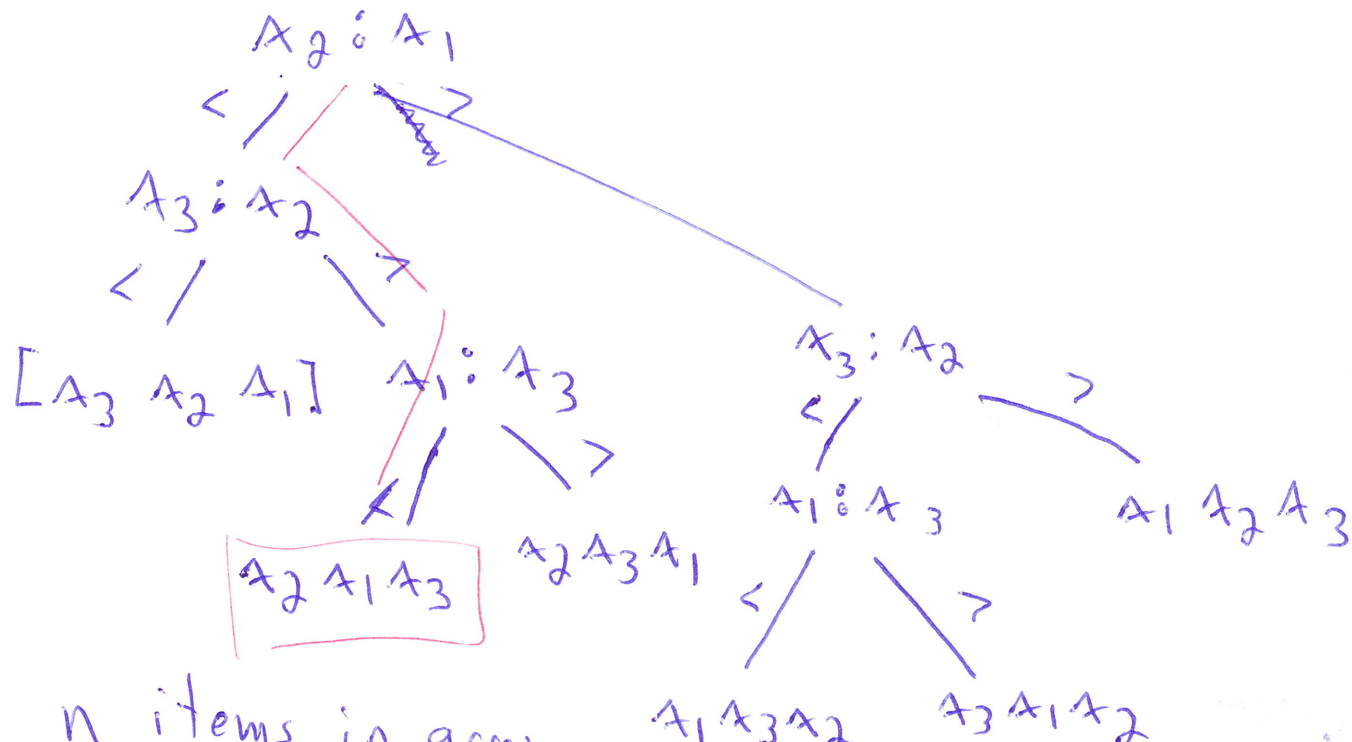
Comparisons: <, >, =

Decision Tree

eg $n = 3$ $[x_1 \ x_2 \ x_3]$

[1 2 3]
[2 1 3]
[3 1 2]
[1 3 2]
[2 3 1]
[3 2 1]

$7! = 5040$

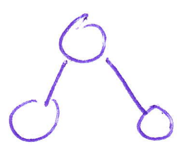


n items in array
leaves: $n!$
internal nodes: # leaves -1
comparisons to find ordering = height of tree

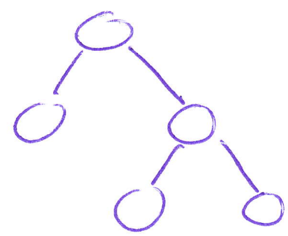
Decision Tree

- use comparisons to distinguish between outcomes
- $n = 3$, 3 items in array
- $n! = 3! = 6$ leafs
- comparisons are $<$, $\geq$
  $\Rightarrow$ each node of tree has exactly 2 children

1 leaf



1 internal
2 leafs



2 internal
3 leafs

- K leafs
  $\Rightarrow$ K-1 internal nodes

How many internal nodes if each node has 3 children?

leafs or leaves?

$n!$ leaves $\Rightarrow$ $n! - 1$ internal nodes

Total: $2n! - 1$ nodes

- Best case, tree is balanced: $\log(2n! - 1)$

$$\geq \log(n!) = \log(n) + \log(n-1) + \ldots + \log(2) +$$

count half terms $\geq \log(n) + \ldots + \log(\frac{n}{2}) \sim \frac{n}{2}$ terms

use smallest term $\geq \log(\frac{n}{2}) + \ldots + \log(\frac{n}{2}) \sim \frac{n}{2}$ terms

$$= (\tfrac{n}{2}) \log(\tfrac{n}{2}) = \tfrac{n}{2} \log n - \tfrac{n}{2} \in \Omega(n \log n)$$

Tree has lower bound height: $\Omega(n \log n)$

Non-comparison based sorting

"stable" • if items have the same key, original order is maintained

Not all sorting algs are stable

eg Hoare's Partition:

77_1 77_2 44_1 44_2 | 55 Pivot

swap

44_2 44_1 77_2 77_1 55

44_2 44_1 55 77_1 77_2

- why do we want to distinguish between the same number?
  - remember there is associated data with keys
    (key, value) pairs

Radix defines what a "digit" is.

For us, R=10 digit: 0...9
R=2 digit: 0, 1

Example R=4 digits 0,1,2,3

R = 1000_4 (base 4) possible digits: $0 \ldots 333_4$

If $R = 1000_4$ then example on slide 40 would sort by whole number instead of last digit only

eg: ⇒ 021
101
123
210
230
232
320

How many buckets were used? 7

How many buckets allocated? $4^3 = 64$

Bucket List array = size R

Appending elements of A to B = size n

$\Theta(n + R)$ auxiliary space

MSD space: $\Theta(n + R + m)$ — # digits = depth of Recursion

Bigger the Radix, fewer digits

- need more space for larger R (buckets)
- fewer digits ⇒ fewer calls to Bucket sort.

Consider Key in Range [0, 999,999]

R = 10 digits 0..9, m = 6 digits

$\Theta(m(n + R)) \Rightarrow \Theta(6(n + 10))$

$\Theta(mnR)$
$\Rightarrow \Theta(6 \cdot 10 \cdot n)$

R = 1000 digits 0..999, m = 2

$= \Theta(2(n + 1000))$ $\Rightarrow \Theta(2 \cdot 1000 \cdot n)$