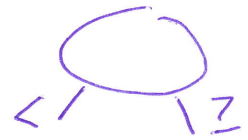
Recall Lower Bound on Comparison-Based Sorting (Decision Tree)

- given array of n numbers
  - $\Rightarrow n!$ permutations (possible outcomes)
  - $\Rightarrow n!$ leaves of decision tree

Aside: Binary Tree: 

- if each node in the binary tree has exactly 2 children $\Rightarrow n! - 1$ internal nodes
  - not needed to show our result but a nice property

Height of binary tree $\geq \log(2(n!)-1) \geq \log(n!)$
$\in \Omega(n \log n)$

Lower Bound on Comparison-Based Search

This time: search for k in items $a_1, a_2, \ldots, a_n$

Possible Outcomes: n (leaves)
- item found at $a_1, a_2, \ldots, a_n$
- 1 or more $\Rightarrow$ not found

Binary tree where each node has 2 children
$\Rightarrow n$ internal nodes - not needed for result

Height $\geq \log(2n+1) \geq \log(n+1) \in \Omega(\log n)$

Search path: follow path from root to a leaf

Talk about "lower bound on a problem" vs algorithm bounds

Interpolation Search

Recall: Binary Search - always choose middle

Based on values of the array, can we make a better guess?

Consider: 0 1 2 3 ... Big #

What influence does the "Big #" have?

When looking for 3?

Binary Search

$T(n) = c + T(n/2) = c + c + T(n/4) \in \Theta(\log n)$

Interpolation Search

$T(n) = T(\sqrt{n}) + c = c + c + T(\sqrt{\sqrt{n}})$

How many times can we $\sqrt{\ }$ n before hit base?

$n^{\frac{1}{2^k}} \leq 1$ not $n(\frac{1}{2})^k$ - more fun with logs

Let $n = 2^m$ $\quad T(2^m) = T(2^{m/2}) + c$

$S(m) = S(m/2) + c = S(\frac{m}{4}) + c + c \in \Theta(\log m)$

$\log m$ to hit base case

$\Rightarrow n = 2^m \Rightarrow m = \log n$

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

Simultaneously use Interpolation & Binary Search $O(\log n)$

Tries

- store bit strings, $\Sigma = \{0, 1\}$
- prefix-free => all keys are leaf nodes
- add \$ "end-of-word" to guarantee prefix-free

Is $\$ \in \Sigma$? No

1\$0\$\$11\$ is not a valid string.

Runtimes: $O(|x|)$ ~ length of bitstring

Keys are in decimal, tries store bit strings

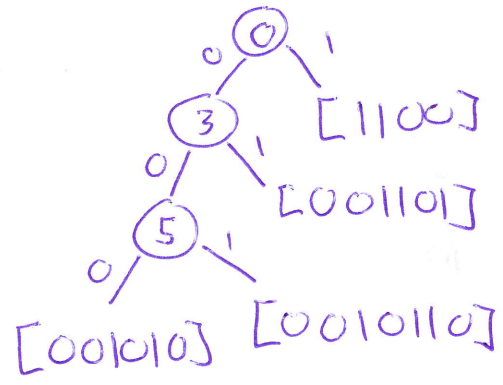
- no leading 0s

k in decimal needs log(k) bits

Int in C: 32 bits, Racket: unbounded

rationals stored $\frac{m}{n}$

Bit	0	1	2	3	4	5	6	7
	0	0	1	1	0	1	\$	
	0	0	1	0	1	0	\$	
	0	0	1	0	1	1	0	\$
	1	1	0	0	\$			



- Find Most-Significant-Bit where they disagree