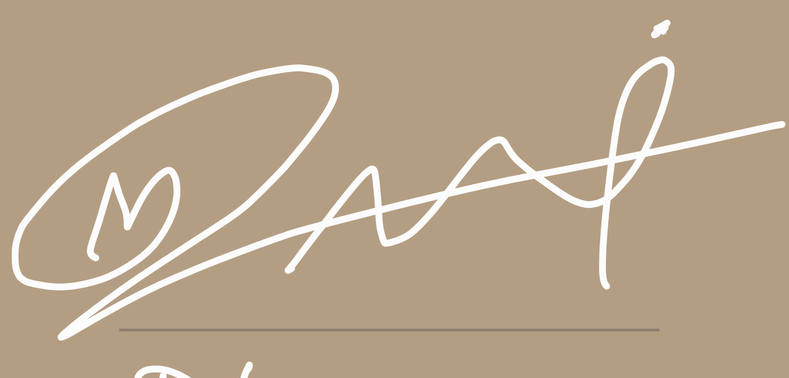
COSC-211 lecture-07

Binary Search

21/09/2021



— Dhyey Mavani

Lecture-07: Data Structures: BINARY SEARCH

→ Simple SSet
→ Binary Search
→ Reasoning about Recursion

Last time: SimpleSSet interface/ADT
— store collection of distinct items that can be compared with $<$ (in code: compareTo method)

Array Implementation: $S = \{x_0, x_1, \ldots, x_{n-1}\}$, $x_0 < \ldots < x_{n-1}$

— store elements in array contents

0	1	2	...	...	...	...	...	n-1	N
x_0	x_1	x_2	.	...	..	...	...	x_{n-1}	

(1) Finding y?

Binary Search: (idea)

— make method to search for y between indices i and k, $(i < k)$.
— compare y to the element x_j, where $\left(j = \frac{i+k}{2}\right)$.
If $y = x_j$, we're good.
If $y < x_j$, look to the left $\equiv (i, j)$
If $y > x_j$, look to the right $\equiv (j, k)$

eg.

	0	1	2	3	4	5	6	7	8
contents:	2	3	5	7	11	13	17	19	

search(12,0,8)
— return index where 12 would be inserted in sorted order.

search(12,0,8)
(i) $j_1 = \frac{0+8}{2} = 4$, look at $x_4 = 11$
$x_4 = 11 < 12$
⇒ look right

search(12,4,8)
(ii) $j_2 = \frac{4+8}{2} = 6$, look at $x_6 = 17$
$x_6 = 17 > 12$
⇒ look left

search(12,4,6)
(iii) $j_3 = \frac{4+6}{2} = 5$, look at $x_5 = 13$
$x_5 = 13 > 12$
⇒ look left

(iv) search(12, 4, 5)
↳ return 5 {because $12 \leq x_5$, $12 \leq 13$}

$\boxed{\text{See the code by Professor}}$

Goal: Classify running time of getIndex with range n as function of n

Denote running time (worst case) $= T(n)$

Question: What is $T(1)$?

$\rightarrow O(1)$ (assuming compareTo is $O(1)$)

Question: What is $T(2)$?

$\rightarrow O(1)$ as well

Question: What is $T(n)$?

$$2^{T(n)} = n \quad \Bigg\| \quad \begin{aligned} T(n) &= O(1) + T\left(\tfrac{n}{2}\right) \rightarrow \text{recurrence relations} \\ T(n) &= O(1) + O(1) + T\left(\tfrac{n}{4}\right) \ldots \end{aligned}$$

Two Math Bits (1) logarithm function
(2) induction

(1) $\log n$ (base 2)

$$\boxed{y = \log_2 x} \text{ iff } \boxed{2^y = x}$$

Properties
(1) $\log(xy) = \log x + \log y$
(2) $\log 2 = 1$
(3) $\log\left(\frac{x}{y}\right) = \log x - \log y$
(4) $\log(x^a) = a \log x$
(5) for every constant $a > 0$, $\log(n) = O(n^a)$

$\Rightarrow$ **claim**: $T(n) = O(\log n)$

$\mathbb{P}$: $\exists\, C > 0, N$ such that $T(n) \leq C \log(n)\ \forall\, n \geq N$

(2) **Induction**: $\mathbb{P}_1, \mathbb{P}_2, \mathbb{P}_3, \ldots, \mathbb{P}_n$ all must hold true.

$\Rightarrow$ we just have to prove $\mathbb{P}_1$ and then assume $\mathbb{P}_K$ is true, then we should prove that $\mathbb{P}_{K+1}$ using $\mathbb{P}_K$ & $\mathbb{P}_1$.

$\Rightarrow$ Whenever $\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_{n-1}$ are true, then $\mathbb{P}_n$ is true

Base Case: starting at $n = N = 2$, $P(1)$ & $P(2)$

$T(2) = O(1)$ (analyzing code)

$\Rightarrow T(2) \leq C_1 = O_1 \cdot \log 2$

So, $P(2)$ is true!

Induction: Show $T(n) \leq C \log(n)$ assuming $T(n') \leq C \cdot \log(n')\ \forall\, n' < n$

$$T(n) = T\left(\tfrac{n}{2}\right) + \underbrace{O(1)}_{C_2} = T\left(\tfrac{n}{2}\right) + C_2 = C \log\left(\tfrac{n}{2}\right) + C_2$$

$$T(n) = C[\log n - \log 2] + C_2 = C \log n + [C_2 - C] = C \log n + C_3 \Rightarrow \boxed{\log(C' \cdot n)}$$

Hence Proved!!