CS1800
Discrete Structures
Fall 2017

Lecture 28
11/9/17

Last time	Today	Next time
· Finish Markov Chains · Start Algorithmic Analysis	- Continue Algorithmic Analysis - Search - Sort	· Sequences & Series

Algorithms for Search

- linear search ⇒ go through elements (pages) one at a time, in order, to find what you want
- Chunk search ⇒ examine chunks of size x to find correct chunk, then perform linear search w/i chunk.
- Binary search ⇒
  - divide elements (pages) in half
  - determine if element you want is in first or second half
  - repeat on correct half.

Analysis

① focus on algorithm's performance as a function of size of input

② worst-case — n

average-case $\frac{1+2+3+\dots+n}{n} = \frac{n(n+1)}{2} \cdot \frac{1}{n} = \frac{n+1}{2}$

best-case — 1

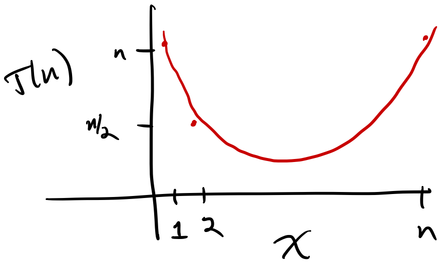
e.g. linear search

linear search

$T(n) = n$

Chunk search

- n pages in total
- chunks of size x



worst case: $T(n) = \frac{n}{x} + x$

$$f(x) = \frac{n}{x} + x$$

$$= n \cdot x^{-1} + x$$

$$f'(x) = -n x^{-2} + 1$$

$$-n x^{-2} + 1 = 0$$

$$1 = n \cdot x^{-2}$$

$$x^2 = n$$

$$x = \sqrt{n}$$

$$\frac{n}{x} = x$$

$$\Leftrightarrow \quad n = x^2$$

$$\Leftrightarrow \quad x = \sqrt{n}$$

$$\Rightarrow \quad T(n) = \frac{n}{\sqrt{n}} + \sqrt{n} = \sqrt{n} + \sqrt{n} = 2\sqrt{n}$$

$$\boxed{T(n) = 2\sqrt{n}}$$

Binary Search

k cuts in half	size of book
0	n
1	$n/2$
2	$n/4 = n/2^2$
3	$n/8 = n/2^3$
4	$n/16 = n/2^4$
⋮	
k	$n/2^k$

What size k yields just one page?

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$k = \log_2 n$$

$$T(n) = \log_2 n$$

n vs. $2\sqrt{n}$ vs. $\log_2 n$

l.s. $T(n) = n$

c.s. $T(n) = 2\sqrt{n}$

b.s. $T(n) = \log_2 n$

		$n = 1000$		$n = 1,000,000$	
L.S.	n	1000	15.75x (L.S. → C.S.); 100x (L.S. → B.S.)	1 000 000	500x (L.S. → C.S.); 50,000x (L.S. → B.S.)
C.S.	$2\sqrt{n}$	64	6.4x (C.S. → B.S.)	2000	100x (C.S. → B.S.)
B.S	$\log_2 n$	10		20	

- Fastest supercomputer in US

  Cray XK7 "titan" Oak Ridge Nat. Lab

  17.59 petaflop

  17.59×10^{15} floating point op. per sec.

- Mac Pro 7 teraflops

  7×10^{12}

$$\frac{17.59 \times 10^{15}}{7 \times 10^{12}} = 2,513x$$

$n = 1,000,000$

l.s. on a supercomputer is 20x slower than B.S. on a Mac.

Sorting

Insertion Sort : $1 + 2 + 3 + 4 + 5 + \dots + n = \frac{n(n+1)}{2} = \frac{n^2}{2} + n/2$

Selection Sort : $n + (n-1) + (n-2) + \dots + 3 + 2 + 1 = \frac{n(n+1)}{2}$

Merge Sort :

6 3 7 2 | 4 8 1 5

~~2~~ 3 6 7 ~~1~~ 4 5 8

1 2 3 4 5 6 7 8

$$T(n) = 2 \cdot T(n/2) + n$$

$$\Rightarrow n \log_2 n$$