Cmp 338

10.1.09

On an Array

Bubble Sort $-\Theta(n^2)$ ← O notation

↑ n^2 steps, if the amount of date is n

Quick Sort $-\Theta(n^2)$ — $O(n\log n)$ on average

merege Sort $-\Theta(n\log n)$ ← not Inplace algrorimen

↑ need double the memory to run vs Quick Sort

Insertion Sort $-\Theta(n^2)$

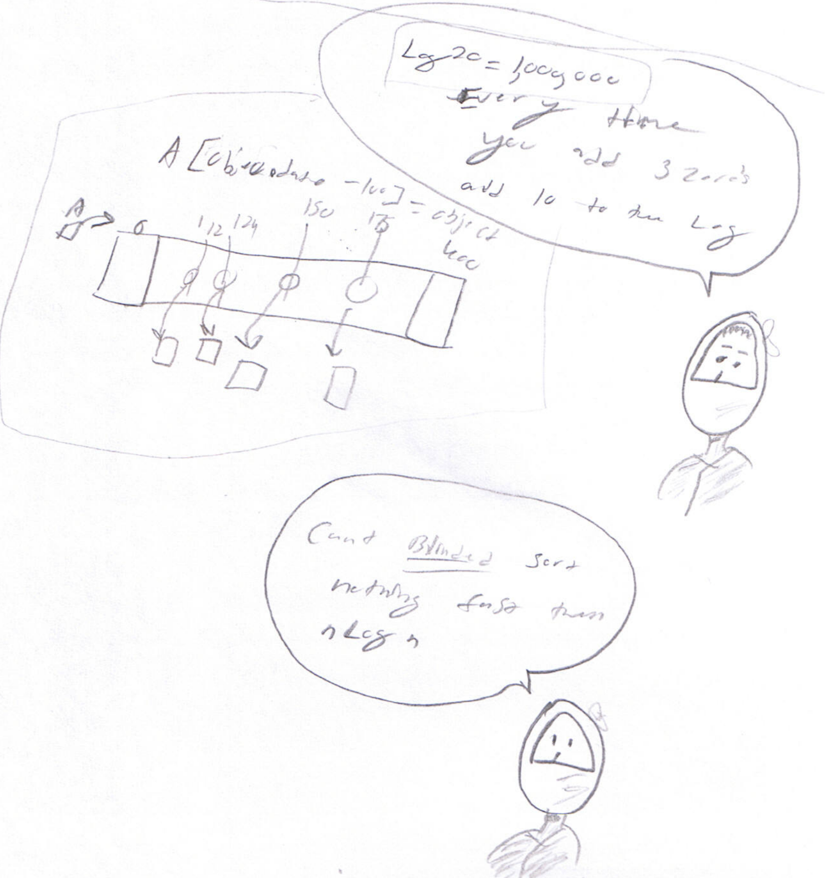
Selection · sort $-\Theta(n^2)$

Linear search $-\Theta(n)$

Binary search $-\Theta(\log n)$ ← fast

~~100~~ $n\log n + 500$~~6~~ $=\Theta(n\log n)$

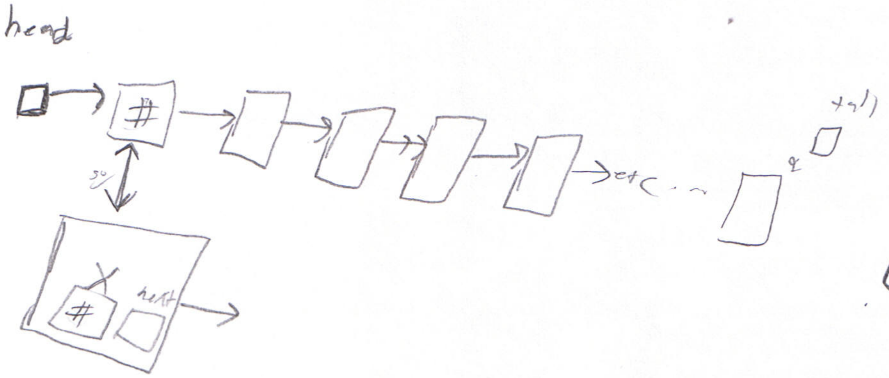
$n\log n - 20$~~00~~ $=\Theta(n\log n)$


A [Objectadata - 100] = object loc
A
0
1/12
124
150
175
Log20 = 1,000,000
Every time you add 3 zero's add 10 to the Log
Cant Blinded Sort nothing faster than nLog n

Cmp 338

2

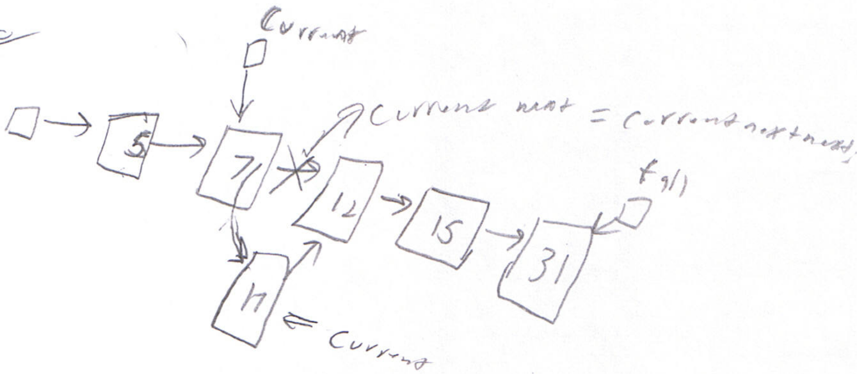
10/1/09



Random access w/Array's

not true w/Linked List

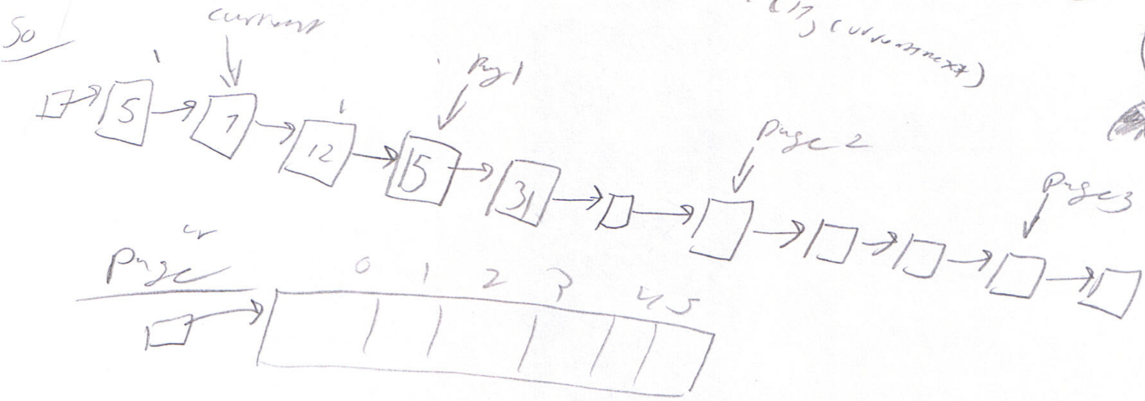
So



current.next = current.next.next;

current.next = new node (17, current.next)

So



Page[i]

Cmp 338

10-1-07

List

← keep sorted

← don't need to keep sorted

	Array	Linked List	Binary search tree
Insert	$\Theta(n)$	$\Theta(1)$	$\Theta(\log n)$
Remove	$\Theta(n)$	$\Theta(1)$	$\Theta(\log n)$
Sort	$\Theta(n\log n)$	$\Theta(n\log n)$	$\Theta(n)$
Search	$\Theta(n\log n)$	$\Theta(n)$	$\Theta(\log n)$

$$\underbrace{\Theta(\log n)}_{\text{search}} + \underbrace{\Theta(n)}_{\text{copy}} = \Theta(n)$$

$\cancel{c_1} \log n + \cancel{\text{smaller terms}} + \cancel{c_2} n + \cancel{\text{smaller terms}}$

$\cancel{c_1 \log n} + \cancel{c_2} n + \cancel{\text{smaller terms}} + \cancel{\text{smaller terms}}$