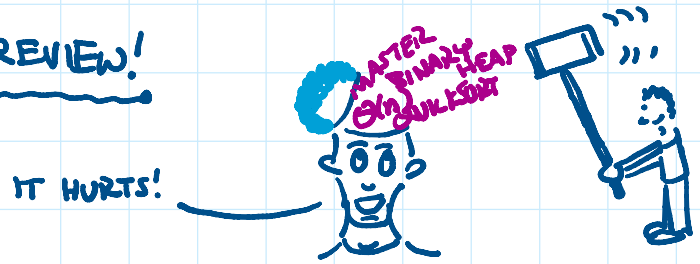
CMSC351-2023-11-18
Wednesday, October 18, 2023 8:49 AM

EXAM 2 REVIEW!



INSERTION SORT

How does it work?
Trace examples
Pseudocode familiarity
After i iterations, leftmost i+1 are correct relative to one another

Tc: Bc $\Theta(n)$ ← sorted list
Wc $\Theta(n^2)$ ← rev. sorted list
Ac $\Theta(n^2)$ ← via inversions!

T/F: Ins. Sort is Wc $O(n^5)$ $\Omega(1)$

Aux Space $\Theta(1)$
In-Place - Yes!
Stable - yay!

Binary Search

How does it work?
List must be sorted
Bc: $\Theta(1)$
Wc: $\Theta(\lg n)$
Ac: $\Theta(\lg n)$
pseudocode etc.

Rec. Reln

defn: $T(n)$ = ... in terms of earlier values w/ base case(s)
examples
Calc. values with
How do these show up in algorithms?

Digging Down

Start w/ $T(n)$ = RR
Sub. in, keep going
Look for a pattern: involve use of k. ← could involve Σ

Sub. in, keep going
look for a pattern: involve var k. ← could involve Σ
When does it end? $k =$ [redacted] via base case
Plug that k back into the patton
to get $T(n) =$ nice expression ← could involve Σ

Trees representing RR

Draw some trees
fill in values

ex $T(n) = 3T(n/2) + 2n^2$
draw the tree representing $T(8)$.

MASTER THM

Works for $T(n) = aT(n/b) + f(n)$

(1) if $f(n) = O(n^c)$ and $\log_b a > c$ then $\Theta(n^{\log_b a})$
(2) if $f(n) = \Theta(n^c)$ and $\log_b a = c$ then $\Theta(n^{\log_b a} \lg n)$
(2f) if $f(n) = \Theta(n^c \lg^k n)$ and $\log_b a = c$ then $\Theta(n^{\log_b a} \lg^{k+1} n)$
(3) if $f(n) = \Omega(n^c)$ and $\log_b a < c$ then $\Theta(f(n))$
Lots of ex!

Merge Sort

How does it work?
Pseudocode ← merging bit!
RR $T(n) = 2T(\frac{n}{2}) + \Theta(n)$
Tc: Bc, Wc, Ac all $\Theta(n \lg n)$ by MT.
Aux Space: $\Theta(n)$ ← via MT!
Stable: yes! ← bc the merge process is stable!
In-place: No!

Heaps, Heapsort

Heaps, Heapsort

Complete binary tree

Facts about node indices and level indices ← nodes 1-indexed, levels 0-indexed

Max heap: key ≥ child keys

maxheapify (aka swapkeydown)

- BC $\Theta(1)$
- WC $\Theta(\lg n)$

Aux space: $\Theta(1)$

Stable: No!

In-Place: Yes!

ConvertToMaxHeap: converts CBT → Max Heap

- BC $\Theta(n)$
- WC $\Theta(n \lg n)$

heapsort - how that works!

- BC: $\Theta(n \lg n)$ if keys all distinct but $\Theta(n)$ if all the same.
- WC: $\Theta(n \lg n)$

Quick Sort

partition fctn - how does it work? What's the result?

- pseudocode

Quicksort uses ↕ and divide and conquer.

RR: $T(n) = T(k) + T(n-k-1) + \Theta(n)$ w/ k = resulting pivot index

TC: WC: $\Theta(n^2)$

BC: $\Theta(n \lg n)$

AC: $\Theta(n \lg n)$

Aux Space $\Theta(1)$

In Place: Yes

Stable: No