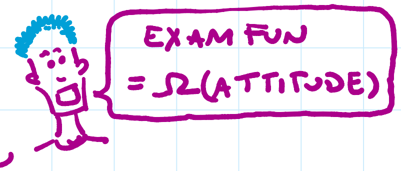
CMSC351-2023-11-16

Monday, October 16, 2023 8:26 AM

REMINDER! EXAM 2 FRIDAY!

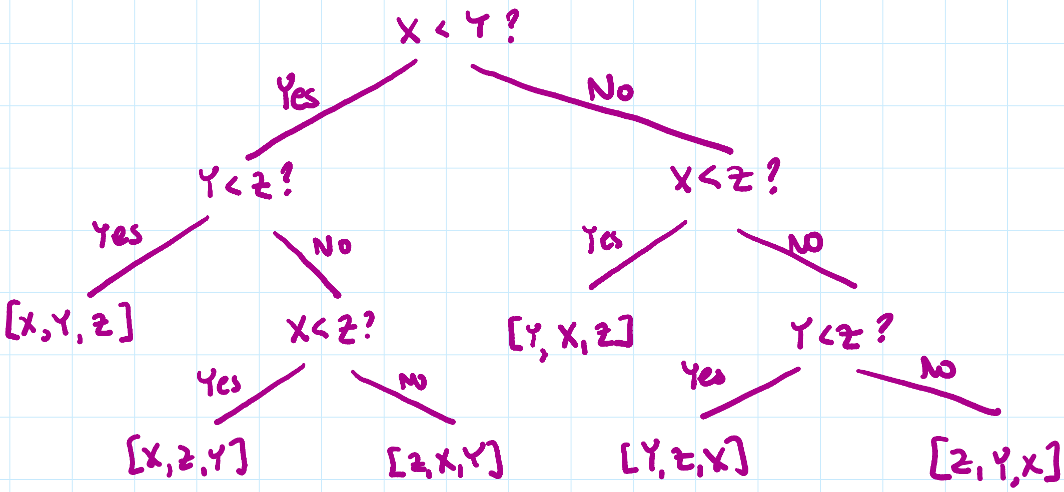
If you can be here and seated by 9am, please do so! :)



Limitations on Comparison-Based Sorting Algorithms, Continued!

① So far: we drew up a table for how bubble sort works on a list $[x, y, z]$ of length $n = 3$.

From this we can draw the corresponding decision tree:



② Two notes

(A) as n gets larger, it seems reasonable that the tree gets larger (wider, higher).

(B) every comparison-based sorting alg. has a corresponding decision tree for each n.

③ Lower Bound on # of decisions in worst-case

(A) Thm: $\Theta(\lg(n!)) = \Theta(n \lg n)$

means: $f(n) = \Theta(\lg(n!))$ iff $f(n) = \Theta(n \lg n)$

Pf: In Later notes!

(B) <u>Obs</u>: For a list of length n there are $n!$ possible sortings and each of those would be a leaf in the corresponding decision tree.

(C) <u>Obs</u>: For a binary tree w/ height h, there are <u>at most</u> 2^h leaves - if the tree is perfect.

(D) Put (B) and (C) together to get.

$$\#\text{leaves in dec. tree} = n! \leq 2^h$$

So now

$$h \geq \lg(n!)$$

Also, each decision is based on a comparison and so we know that in the <u>worst</u> case

$$\#\text{decisions} \geq \lg(n!)$$

Consequently, since each comparison/decision takes constant time we then have

$$\text{time} \geq \lg(n!)$$

So

$$T(n) = \Omega(\lg(n!)) \quad \circledast$$

then apply the theorem (essentially) to get

$$T(n) = \Omega(n \lg n).$$

<u>THUS</u>, in conclusion: In the worst-case we have

$$T(n) = \Omega(n \lg n)$$

(E) <u>Meaning</u>: Reflect on our comparison-based alg:

Some have same B-Case and W-C:

<u>ex</u> Heap Sort: both $\Theta(n \lg n)$ (all distinct)

ex Heap Sort : both $\Theta(n\lg n)$ (all distinct)

ex Merge Sort : " $\Theta(n\lg n)$

ex Bubble Sort: " $\Theta(n^2)$

Some have different

ex Insertion Sort: BC $\Theta(n)$ WC $\Theta(n^2)$

Interpreting WC: Think "there are one or more annoying lists!"

ex When we say insertion sort is $\Theta(n^2)$ WC

we mean: For any n there are some annoying lists which take time $\Theta(n^2)$.

Focus on the fact that this is $\Omega(n^2)$.

Conclusion: It doesn't matter how you come up with a comparison-based sorting alg.

There will always be one or more annoying lists which will take $\Omega(n\lg n)$ time.

Question: Could we come up w/ a sorting alg which is not based upon comparisons and for which every list can be done in under $\Omega(n\lg n)$ time?

Answer: Yes! Counting Sort can do $\Theta(n)$ under certain restrictions.

Note: In ⊛ we don't quite use the thm.

rather we use a part of it which proves:

if $f(n)=\Omega(\lg(n!))$ then $f(n)=\Omega(n\lg n)$