Sorting & Algorithmic Bounds, Lec# 25

Sorting is a useful task.
- Can be used to solve not-so-obvious tasks.
  - Sorting improves duplicate finding.
  - 3SUM problem (triplet of vals in an array add to 0).

How hard is it to sort?

Can we do better than $n \log n$?

Math problems:

Is $N! \in \Omega((N/2)^{N/2})$?

$N!: (N)(N-1)(N-2)(N-3)\cdots(N-N)$

$(\frac{N}{2})^{\frac{N}{2}}: (\frac{N}{2})(\frac{N}{2})(\frac{N}{2})\cdots$ ($\frac{N}{2}$ times.)

$N = 10$:

$10! = 10 \times 9 \times 8 \times \cdots 1$

$5^5 = 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5$

$10!$ is obviously larger than $N!$.

$N!$ is atleast large as $(\frac{N}{2})^{\frac{N}{2}}$

Given that $N! > (\frac{N}{2})^{\frac{N}{2}}$

Show $\log(N!) \in \Omega(N \log N)$

$\log(N!) > \log(\frac{N}{2})^{\frac{N}{2}}$

$\log(N!) > \frac{N}{2}\log(\frac{N}{2})$

$\log(N!) > N \log(N)$

Show that $N \log N \in \Omega(\log(N!))$

$\log(N!) = \log(N) + \log(N-1) \cdots + \log(1)$

$N \log N = \log(N) + \log(N) + \cdots + \log(N)$ (bigger than $\log(N!)$)

Given:

$N \log N \in \Omega(\log(N!))$

$\log(N!) \in \Omega(N \log(N))$

Informally:

$N \log N \geq \log(N!)$

$\log(N!) \geq N \log N$

We can say:

$N \log N \in \Theta(\log N!)$

$\log(N!) \in \Theta(N \log N)$

Informally:

$N \log N = \log(N!)$

Many sorting algorithms do $\Theta(N \log N)$ worst case time.
Can we do better?

Let TUCS (ultimate comparison sort) be the asymptotically fastest possible comparison sorting algorithm. Let $R(N)$ be its worst case runtime in Θ notation.

- Worst case RT of TUCS has to be $O(N \log N)$. Why?
  - TUCS needs to be the fastest. If it was worse, then Mergesort would be better
  - ~~Worst case runtime is $\Omega(1)$~~
  - ~~There is no~~
- Worst case RT is $\Omega(N)$
  - Have to look at every item.

We will prove that it's impossible to be better than $\Omega(N \log N)$.

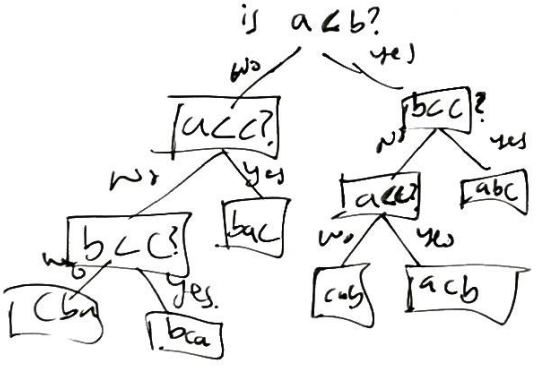
Puppy, Cat, Dog:

We have a, b, c. Put in sorted order.

a<b	b<c	Which is Which?
yes	yes	a: puppy b: cat c: dog (abc)
no	no	a: dog b: cat c: puppy (cba)
yes	no	-> 2 options

a) a: puppy b: dog c: cat (acb) we have to
b) c: puppy b: dog a: cat (cab) ask another q.
a < c?

Decision tree:



How many questions do you need if you have 4 items?
5!
We have a binary tree w/ 24 leaves (4!)
- Min levels: $\log_2(24) = 4.5 \Rightarrow 5$.

How many questions do you need to solve for N items?

→ $\Omega(\log(N!))$

Decision tree needs N! leaves.

We need $\log(N!)$ levels which is $\Omega(\log(N!))$

Deriving questions is hard.

→ So just sort them!

leftmost = puppy

middle = cat

right = dog.

- Puppy Cat Dog reduces to sorting

→ any lower bound on puppy cat dog also applies to sorting.

- Sorting also takes $\Omega(\log(N!))$

- So MCS needs at least $\Omega(\log(N!)) = N\log N$.

Mathematically impossible to sort using fewer comparisons.