

COP 3503 9/24/28

⑥ Whiteboard Sessions ✓

① Quick Select Analysis ✓

② MCSS ✓

③ Lower bound comparison sort

→ Run Partition - time n w/ prob 1

Prob $\frac{1}{n}$ Split $0, n-1 \Rightarrow \frac{1}{n}(1) + \frac{0}{n}\tau(0) + \frac{n-1}{n}\tau(n-1)$

Split $k, n-1-k \Rightarrow \frac{1}{n}(1) + \frac{k}{n}\tau(k) + \frac{n-1-k}{n}\tau(n-1-k)$

...

$$\tau(n) = n + \frac{1}{n} \left[\begin{array}{l} \frac{1}{n} + \frac{0}{n}\tau(0) + \frac{n-1}{n}\tau(n-1) \\ \frac{1}{n} + \frac{1}{n}\tau(1) + \frac{n-2}{n}\tau(n-2) \\ \dots \end{array} \right]$$

$$\frac{1}{n} + \frac{n-1}{n}\tau(n-1) + \frac{0}{n}\tau(0)$$

$$\tau(n) = n + 1 + \frac{1}{n} \left[\sum_{i=0}^{n-1} \frac{i\tau(i)}{n} \right]$$

$$\tau(n) = (n+1) + \frac{1}{n^2} \sum_{i=0}^{n-1} i\tau(i) \cdot 2 \leftarrow \begin{array}{l} \text{error} \\ \text{from} \\ \text{class} \end{array}$$

$$n^2 T(n) = n^2(n+1) + \sum_{i=0}^{n-1} 2i T(i) \quad \begin{array}{l} \nearrow \text{error} \\ \nearrow \text{from} \\ \nearrow \text{class} \end{array} \quad \begin{array}{l} n^3 + n^2 \\ - (n^3 - 2n^2 + n) \end{array}$$

$$-(n-1)^2 T(n-1) = (n-1)^2(n) + \sum_{i=0}^{n-2} 2i T(i)$$

error from class!!!

$$\underline{n^2 T(n) - (n-1)^2 T(n-1)} + \underline{3n^2 - n + 2(n-1) T(n-1)}$$

$$n^2 T(n) = (n-1)(n-1+2) T(n-1) + 3n^2 - n \quad \nearrow \text{error from class}$$

$$\frac{n^2 T(n)}{n(n+1)} = \frac{(n-1) T(n-1)}{n(n+1)} + \frac{3n^2 - n}{n(n+1)}$$

$$\frac{n T(n)}{n+1} = \frac{(n-1) T(n-1)}{n} + \frac{3n^2 - n}{n(n+1)}$$

fixed!

$$\frac{n T(n)}{n+1} \leq \frac{(n-1) T(n-1)}{n} + \frac{(3n^2 - n)}{n(n+1)}$$

$$\text{Let } S(n) = \frac{n T(n)}{n+1}$$

$$S(n) \leq S(n-1) + \frac{3n^2 - n}{n(n+1)}$$

$$S(n) \leq S(n-1) + 3$$

$$S(n) \leq S(n-2) + 3 + 3$$

$$\dots \quad S(n) \leq S(0) + \underbrace{3+3+3+\dots}_{n \text{ times}} = 3n$$

$$3n = \frac{n \tau(n)}{n+1}$$

$$\tau(n) = \frac{3n(n+1)}{n} = 3(n+1) = O(n)$$

Maximum Contiguous Subsequence Sum

7, -3, 2, -6, 3, -8, 9, 10, 2, -4, 3, 2, -4

let $n = \text{length of array}$

Brute Force

loop over all $[i, j]$, $0 \leq i \leq j \leq n-1$
for each range do a loop to add vals.

① for i

② for j

sum = 0

③ for k

sum += a[k]

$O(n^3)$

sum = 6

known $i=0$ $j=2$

figure out sum $i=0$ to $j=3$?

for i

sum = 0

for j

sum += a[j]

$O(n^2)$

-5

7, -3, 2, -6, 3, -8, 9, 10, 2, -4, 3, 2, -4

prefix
sum

7 4 6 0 3 -5

NEG			X _i	X _{i+1}	

Does it ever make sense to prepend a negative prefix? - NO!

P P P P P N

N P P

```
int sum = 0, res = a[0];
```

```
for (int i = 0; i < n; i++) {
```

```
    sum += a[i];
```

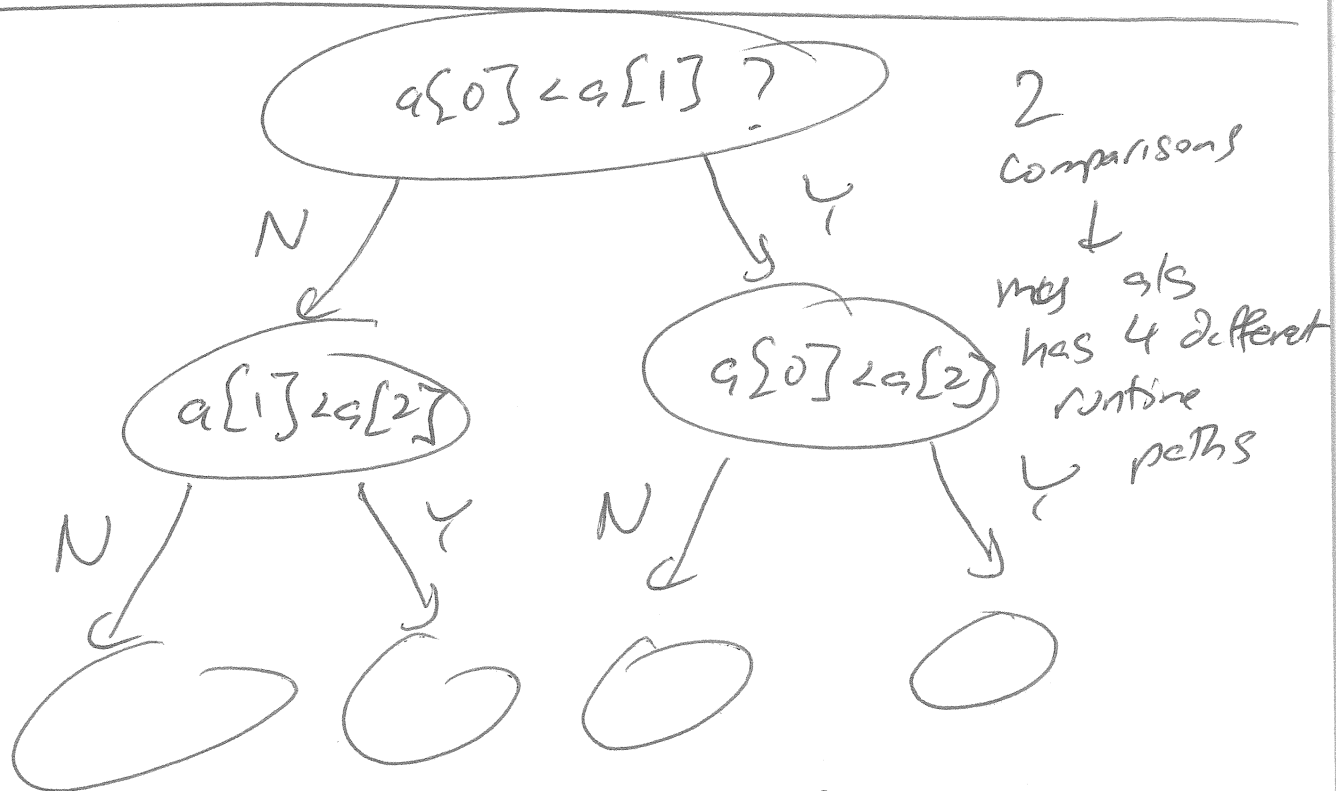
```
    res = max(res, sum);
```

```
    if (sum < 0) sum = 0;
```

```
}
```

$O(n)$

Lower bound for comparison sorting



Sorting 3 items $\rightarrow$ any^{or} of $\bullet 3! = 6$ orders.

$\left. \begin{array}{l} 1 \ 3 \ 2 \\ 2 \ 1 \ 3 \end{array} \right\}$ need a different set of swaps

Alg has to have ≥ 6 diff run paths

$t_c =$ run time (# decisions)

$$\begin{array}{ccc}
 2^{t_c} & \geq & n! \\
 \downarrow & & \downarrow \\
 \text{max run paths} & & \# \text{ of unique inputs}
 \end{array}$$

Stirling's Approx for $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

$$2^k \geq \left(\frac{n}{e}\right)^n$$

$$\log_2 2^k \geq \log_2 \left(\frac{n}{e}\right)^n$$

$$k \geq n \log_2 \left(\frac{n}{e}\right)$$

$$k \geq \underline{\underline{n(\log_2 n - \log_2 e)}}$$

$$= O(n \lg n)$$

$$= \Omega(n \lg n) \quad \begin{array}{l} \text{at least} \\ n \lg n \end{array}$$

Bucket Sort, Counting Sort, Radix Sort

Bucket Sort

Only works if values are relatively uniformly distributed

n values

range $[S, E)$

let ~~or buckets~~ sz
create n buckets, each of size $= \frac{E-S}{n}$.

0 $[S, S+sz) \rightarrow$

1 $[S+sz, S+2sz) \rightarrow$

2 $[S+2sz, S+3sz) \rightarrow$

$n=10$ $S=3$, $E=5$

(4.12), 3.99, 3.1, 4.2, 3.7, 3.6, 4.9, 4.99, 4.43, 3.72

0 $[3, 3.2) \rightarrow 3.1$

1 $[3.2, 3.4)$

2 $\rightarrow 3.6, 3.7, 3.72$

3 $\rightarrow 3.99$

4 $\rightarrow 4.12$

5 $\rightarrow 4.2$

6 $\rightarrow 4.43$

7 $\rightarrow 4.9, 4.99$

8

9

Counting Sort

all input values are integers ^{in range} $0, \dots, D-1$

D is relatively small. N integers

$D = 5$

$N = 13$

List

(0, 2, 2, 4, 3, 1, 2, 3, 3, 1, 0, 0, 4)

~~freq array~~

~~3 2 3 3 2~~
0 1 2 3 4

cum freq array of

~~3 5 8 4 8~~
2 4 7 10 12
+ 3 6 9 11
0 2 5 8 10

Sorted

0₁ 0₂ 0₃ 1₁ 1₂ 2₁ 2₂ 2₃ 3₁ 3₂ 3₃ 4₁ 4₂
0 1 2 3 4 5 6 7 8 9 10 11 12

for ($i = n-1; i \geq 0; i--$) {

STABLE
SORT

sorted[~~i~~] = list[i]

cf[list[i]]

cf[list[i]]--;

MERGE
SORT
STABLE

TIES IN OLD
LIST STAY IN
THE SAME

RELATIVE ORDER

QUICK
SORT
IS NOT