

COP 3502 1/27/26

1) Experimental Run Time

2) Practice Alg Analysis Qs.

3) Recursion Intro

① general analysis

② Timing

③ Sum/Rec Rel

→ Have ~~code~~ code, run it for several diff input sizes determine likely Big-O runtime.

20	.7 s
21	.85 s
22	1.12 s
23	2.29 s
24	3.06 s
25	5.75 s
26	10.78
27	21.6
28	76.9

FIND CI Fall 25

$1 \ll k$ is 2^k

$hi-lo$ starts as 2^k

each while loop iter divide $hi-lo$ by 2

How many times do I divide 2^k by 2 to arrive at 1? \boxed{k}

Inside while loop we do $O(n)$ work for each iteration. Total runtime $O(nk)$.

FIND CI Sum 25

0000
0001
0010
⋮

LSB flips every time $2^n - 1$ times
2nd least sig bit = = $2^{n-1} - 1$ times

1111
↑

MSB = $2^1 - 1$ time

0000
0000
0010
0010

$$f(n) = \sum_{i=1}^n 2^i - 1$$

$$= \sum_{i=1}^n 2^i - \sum_{i=1}^n 1$$
$$= \sum_{i=0}^n 2^i - 2^0 - n$$

$$= \frac{2^{n+1} - 1}{2 - 1} - n - 1 = \boxed{2^{n+1} - n - 2}$$

End CI Fall 24

Alg analysis

Step 1 - $O(n^2)$ time

Step 2 \rightarrow also $n \times O(n) = O(n^2)$ time
make n heaps of size n

Step 3 $\rightarrow n \times O(\lg n) = O(n \lg n)$, delete min
takes $O(\lg n)$ time.

Step 4 repeats n^2 times

a. $O(n)$ max list length = n

b. $O(1)$ add to end

c. $O(\lg n)$ heap ~~is~~ delete

\hookrightarrow dominated by $O(n)$

\Rightarrow Total run time = $O(n^3)$

END CI Fall 23

How many strings of length n contain As + Bs only?

AAA BAA

AAB BAB

ABA BBA

ABB BBB

$n=3$

$$\underbrace{2 \times 2 \times 2 \times 2}_{2^n} = 2^n$$

go through each letter, $O(1)$ per letter
 $O(n)$ $\frac{n}{n}$
 $O(n2^n)$

Recursion is simply a function that calls itself. Here is a famous example

```
int fact (int n) {  
    if (n <= 1) return 1;  
    return n * fact(n-1);  
}
```

Factorial for 5 fact(5)

