COP3502 1/18/24

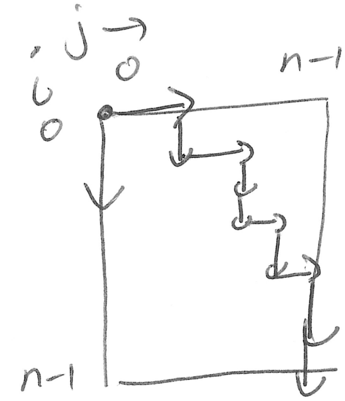
look at some more code segments

recursion → Recurrence Relations

Analysis Problems - past fnd.

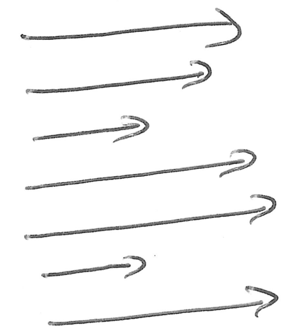
```
i=0, j=0;
while (i<n) {
    while (j<n && array[i][j]==1)
        j++;
    i++;
}
```

O(n)



```
i=0;
while (i<n) {
    j=0;
    while (j<n && array[i][j]==1)
        j++;
    i++;
}
```

$O(n^2)$



```
int* isprime = calloc(n+1, sizeof(int));
for (int i=2; i<=n; i++) isprime[i]=1;
for (int i=2; i<=n; i++)
    for (int j=2*i; j<=n; j+=i)
        isprime[j] = 0;
return isprime;
```

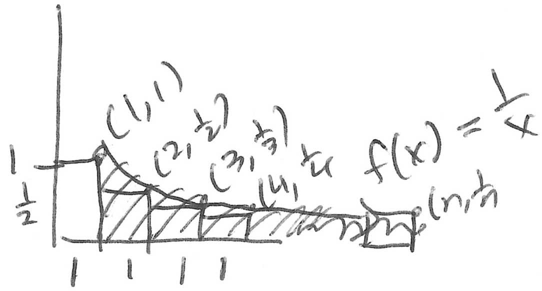
~~1~~	2	3	~~4~~	5
~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~
~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~

$$\frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \cdots + \frac{n}{n}$$

$$H_n = \sum_{i=1}^{n} \frac{1}{i}$$

$$= \sum_{i=2}^{n} \frac{n}{i} = n \sum_{i=2}^{n} \frac{1}{i} \le n \ln n$$

$$O(n \lg n)$$



$$\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} \le \int_1^n \frac{1}{x}\,dx$$

$$= \ln x \Big|_1^n$$

$$= \boxed{\ln n}$$

0														n−1
G	G	G	G	G	G	G ...	G	X	X	X	X	X	X	X

n boxes
ask does box k have a gold coin?
after getting 2 NO responses must correct ans
total # of gold coins.
Run-time = # Questions you ask.

Strategy 0: Go every box in order $O(n)$

Strategy 1: Go every other $\to O(n)$

Strategy 2-4: Also $O(n)$ (1^{st} guess $n-1, \frac{n}{2}, \frac{n}{8} \ldots$)

Strategy 5: 1,3,6,10 (1-based) Triangular #s.

most yes = k

$$\frac{k(k+1)}{2} = n$$

$$k(k+1) = 2n$$

$$k^2 + k - 2n = 0$$

$$k = \frac{-1 + \sqrt{1^2 + 4(1)2n}}{2} \sim \frac{\sqrt{8n}}{2} = O(\sqrt{n})$$

until 1st no

$\frac{k(k-1)}{2}$ to $\frac{k(k+1)}{2}$ diff k

$O(\sqrt{n})$ before 2nd no

$$\boxed{O(\sqrt{n})}$$

Guess: $\sqrt{n}$, $2\sqrt{n}$, $3\sqrt{n}$, ... $\sqrt{n} \times \sqrt{n}$

y y y n

$\sqrt{n}$ multiples $\sqrt{n}$

$\sqrt{n}$ more

$O(\sqrt{n})$

Sqrt Decomposition

$n! = (1 \times 2 \times 3 \times \cdots (n-1)) \times n$

$\boxed{n! = (n-1)! \times n}$ $\boxed{0! = 1}$

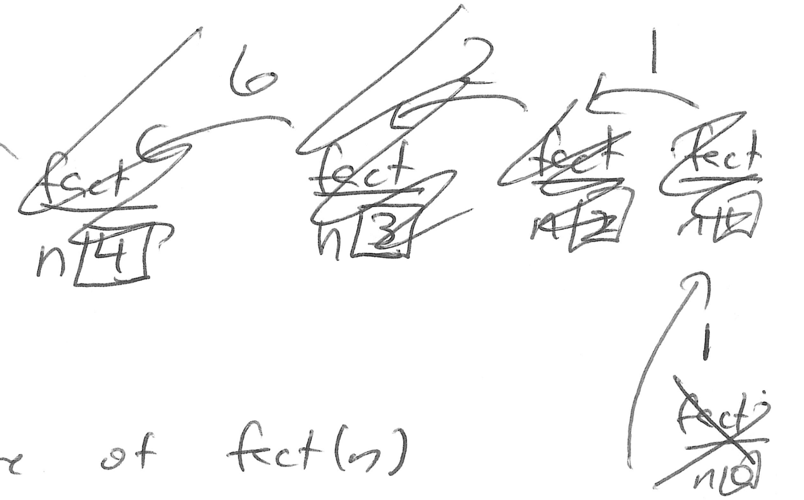
```
int fact(int n) {
    if (n==0) return 1;
    return n * fact(n-1);
}
```

~~fact(0)~~
~~fact(1)~~
~~fact(2)~~
~~fact(3)~~
~~fact(4)~~
main

main
int x = fact(4)

24

$x\ \boxed{24}$



let $T(n)$ = run time of fact(n)

1. $O(1)$
2. $T(n-1)$

$T(n) = O(1) + T(n-1)$

$T(n) = \boxed{T(n-1) + 1}$ 1st iter $\quad T(n-1) = T(n-2) + 1$

$T(n-2) = T(n-3) + 1$

$= T(n-2) + 1 + 1$

$= \boxed{T(n-2) + 2}$ 2nd iter

$= T(n-3) + 1 + 2$

$= \boxed{T(n-3) + 3}$ 3rd iter

After k iterations we have

$T(n) = T(n-k) + k$

known data pt

$\boxed{T(0) = 1}$

Let $k = n$ and plug in (let $n-k=0$ because I know $T(0)$. $\Rightarrow k = n$.)

$T(n) = T(n-n) + n$

$= T(0) + n$

$= \boxed{1 + n}$ $\quad O(n)$

Fall 2023 C1

n, k - input
gen all strings AsBs of length n
for each count if it ever has at least k Bs in a row.

run-time going through 1 string is $O(n)$.
# of strings — 2^n

$O(n * 2^n)$

n=3
AAA
AAB
ABA
ABB
BAA
BAB
BBA
BBB
8

Summer 2023 C1

1st 3 lines = $O(n)$ mem
4th line = $T(\frac{n}{2})$ memory

$T(n) = T(\frac{n}{2}) + n$ 1st
$= T(\frac{n}{4}) + (\frac{n}{2} + n)$ 2nd
$= T(\frac{n}{8}) + (n + \frac{n}{2} + \frac{n}{4})$ 3rd

$T(\frac{n}{2}) = T(\frac{n}{4}) + \frac{n}{2}$
$T(\frac{n}{4}) = T(\frac{n}{8}) + \frac{n}{4}$

$T(1) = 1$

After k iterations

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

let $\frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow k = \log_2 n$

$$T(n) = T(1) + \sum_{i=0}^{\log_2 n - 1} \frac{n}{2^i}$$

$$\leq T(1) + \sum_{i=0}^{\infty} \frac{n}{2^i} = 1 + n(2) = \boxed{O(n)}$$

Sum $= \frac{1}{1-\frac{1}{2}} = 2$

Spring 2023

$O(\max(M,N))$

loop runs M or N times, whichever it larger due to the OR.

Fall 2021

Let $T(n)$ = run time of recursive function on array size n.

$$T(n) = 1 + T(n-1) + 1 + T(n-1) + 1 + T(n-1)$$

$$T(n) = \boxed{3T(n-1) + 1}, \quad T(1) = 1$$

$T(n-1) = 3T(n-2)+1$

$$= 3\left(3T(n-2)+1\right)+1$$

$$= \boxed{9T(n-2) + (3+1)}$$

$$= 9\left(3T(n-3)+1\right) + (3+1)$$

$$= \boxed{27T(n-3) + (9+3+1)}$$

After k iterations, we have

$$= 3^k T(n-k) + \sum_{i=0}^{k-1} 3^i$$

let $n-k=1 \Rightarrow k=n-1$

$$T(n) = 3^{n-1} T(n-(n-1)) + \sum_{i=0}^{n-2} 3^i$$

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

$$\cong \frac{3}{2}\left(3^{n-1}\right) = O(3^n)$$

Summer 2020

find n. Ask is there a room 2^k spots forward?
Minimize worst case # questions.

Ask $2^0, 2^1, 2^2, 2^3, \ldots\ 2^m$ (1st no)

$n \geq 2^{m}, n < 2^{m+1}$

Ask $2^{m-1}, 2^{m-2}, 2^{m-3} \ldots\ 2^0$

$\boxed{1}\ \boxed{2}\ \boxed{4}\ \boxed{8}\ \boxed{16}\ \boxed{32}\ \boxed{64}\ \boxed{128}\ \boxed{144}\ \boxed{152}\ \boxed{156}$

128 n
64 n
32 n
16 y
8 y
4 y
2 n
1 n

$2^{m+1} > n > 2^{m-1}$

runtime $= O(m)$

$2^{m-1} \leq n$

$m-1 \leq \log_2 n$

$m \leq 1 + \lg n$

$m = O(\lg n)$