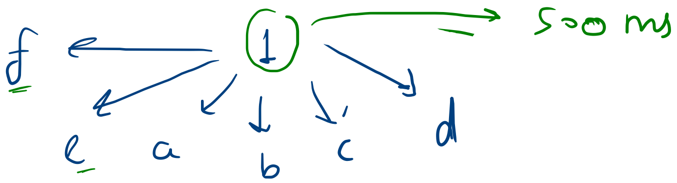
Time & Space Complexity

↳ why?



PC → slow

time ↳

login ↳ local storag

int a=10; → 5ms
↘ 50ms

× Calgo forge
↳ login/signup

3 pabG ↳

loops

$k = 5ms$

$k = 50 ms$

$O(1)$ → constant

$n = 1000000$

```
int a = 5;     → O(1)
int b = 6;     → O(1)
int n = 10;    → O(1)

for(int i=0; i<n; i++){     n times
      a = a+b;
}
```

$10^7, 10^8$

$10^7 + 3 \simeq 10^7$

$10^5 + 6$

$10^7 + 6$

$10k + 6k$

$16k$

$n = 10$

total = $\boxed{nk + 6k}$ → $16k$

$100000k + 6k$ → 100006

$100k + 6k$ → $106k$

$1000k + 6k$ → $1006k$

$0.6k$

✓ We don't measure constant operations.

marks = 30

marks = 95

best case

→ 2 op

```
X  if(marks > 90){
          print("Excellent");
X } else if(marks > 80){
          print("Good");
X } else if(marks > 60){
          print("Above Average");
X } else if(marks > 40){
          print("Avergae");
   } else {
✓         print("Try Again");
   }
```

→ 2 operations

ave → 4 op

worst case → 6 operations

window → O(N)

→ 1 sec

mac

→ 0.5 sec

Big O

↓

O notation

→ worst case

Θ notation

→ average case

Ω notation

→ best case

↓

$\Omega(n^2)$

worst case ← O(N)

O(1)

O(logN)

Θ(n)

```
int a = 5;
int b = 6;

int n = 3;

for(int i=1; i<=n*n*n; i++){
      b = a+1;
      a = b;
};

for(int i=1; i<=n*n; i++){
      a = b+1;
      a = b;
      b = 4;
}

for(int i=1; i<=n; i++){
      a = a + b;
}
```

int a = 5; → $O(1)$ → constant

int b = 6; → $O(1)$

int n = 3; → $O(1)$

first loop → $2N^3$

second loop → $3N^2$

third loop → N

$O(N^3)$

$N = 10$

$N = 10^3$

total-time

↳ $\underbrace{2N^3}+\underbrace{3N^2}+\boxed{N}+k$

$O(2N^3) \simeq O(N^3)$

what, how, why

Ques

N=5 , 25

n = 5

i = ~~1~~ 2 i<=5

```
for(int i=1; i<=n; i++){

    for(int j=1; j<=n; j++){
            int a = i*j;
    }

}
```

$5+5+5+5+5 \rightarrow 25$

$O(N^2)$

$N+N+N+N+N \rightarrow 5N$

j = ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

$N+N+N+N+N+N \rightarrow 6N$

i = ~~1~~ ~~2~~ ~~3~~ 4 5

→ j = {1, 5}   (N)

→ j = {1, 5}   N

→ j = {1, 1   N

N

N

$N+N+N+N \rightarrow$ N tim

$O(N^2)$

Ques 2

$n=5$

```
for(int i=1; i<=n; i++){

    for(int j=1; j<=i; j++){
        int a = i*j;
    }

}
```

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

$$\overbrace{1+2+3+4+5+6+7+8 - - - n}$$

$$1+2+3+4+5+ - - - - - - +n \rightarrow \frac{n(n+1)}{2}$$

$n=5$
$$\rightarrow \overbrace{1+2+3+4+5}$$

$n=6$
$$\rightarrow 1+2+3+4+5+6$$

21

$n=8$
$$\rightarrow 1+2+3+4+5+6+7+8$$

$$\left(\frac{n^2}{2}+\frac{n}{2}\right) \rightarrow \left(\frac{n^2}{2}\right) \simeq O(N^2)$$

$\boxed{\log_2^N}$

$N \rightarrow 1$

↳ divide by x

↳ $\boxed{\log_x^N}$

⟷

$1 \rightarrow N$

↳ multiply by x

↳ $\boxed{\log_x^N}$

Ques 3

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

$O(\log N)$ ←

$N = 256$

$i = 256$

↳ $i = 128$

↳ = 64

256
↳ 128
↳ 64
↳ 32
↳ 16
↳ 8
↳ 4
↳ 2
↳ 1

$$\underbrace{N \quad \frac{N}{2} \quad \frac{N}{4} \quad \frac{N}{8} \quad \frac{N}{16} \quad \frac{N}{32} \quad \frac{N}{64} \quad \frac{N}{128} \quad \frac{N}{256} \quad \text{----} \quad 1}_{x}$$

$$\hookrightarrow \frac{N}{2^0} \quad \frac{N}{2^1} \quad \frac{N}{2^2} \quad \frac{N}{2^3} \quad \frac{N}{2^4} \quad \frac{N}{2^5} \quad \frac{N}{2^6} \quad \text{----} \quad \frac{N}{2^x}$$

$$\frac{N}{2^x} = 1$$

$$2^x = N$$

$$\ln 2^x = \ln N$$

$$\rightarrow x \ln 2 = \ln N$$

$$x \log_2 2 = \log_2 N$$

$$x * 1 = \log_2^N$$

$$= \boxed{x = \log_2^N}$$

$\ln a^b$

$\rightarrow b \ln a$

$\ln \rightarrow \log_2^x$

$\log_a^a \Rightarrow 1$

Ques

$$\overbrace{1 \to 2 \to 4 \to 8 \to 16 \ldots - - - N}^{\log_2^{N}}$$

```
for (int i = 1; i < n; i++) {
    i *= k;
}
```

$\hookrightarrow \quad O\left(\log_k^{N}\right)$

Ques

$\left\{ \frac{n}{2} \times \log N \right.$

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {

    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }

}
```

$\frac{n}{2} \leftarrow$

$\rightarrow \log_2 N$

$\frac{N \log N}{2} \simeq O(N \log N)$

Ques

$i = n \quad i = \frac{n}{2} \quad i = \frac{n}{4}$

```
fun(int n){
    int k = 0;
    for (int i = n; i > 0; i = i / 2)
    {
        for (int j = 0; j < i; ++j)
        {
            ++k;
        }
    }
    print(k);
}
```

$n = \cancel{64}\ \cancel{32}\ \cancel{16}\ \cancel{8}\ \cancel{4}\ \cancel{2}\ 1\ 0$

$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \frac{N}{16} + \frac{N}{32} + \frac{N}{64}$

$\underbrace{64 + 32 + 16 + 8 + 4 + 2 + 1}$

GP

$a = 1$

$r = 1/2$

$\left\{ \frac{a(r^n - 1)}{r - 1} \right\}$

$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \frac{N}{16} + \frac{N}{32} + \cdots 1$

$N\left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \cdots\right)$

$N\left(1 + 0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 + \right)$

$\simeq 2N$

$O(2N) = O(N)$

Space Complexity

input space

+

{ extra space

time_allowed ⇒ 1s

time-limit ↳ 1s

$1s \simeq 10^8$

$N = 10^5$ ↳ $N\log N$

$O(N)$

$N = 10^6 - 10^7 \to 10^8$ ↳ $O(N)$

$2 \to 8$

$N = (1 - 10^4)$ ↳ $O(N^2)$

$N = 10^9 - 10^{10} \to 10^{12-15}$ ↳ $O(1) \leftrightarrow O(\log N)$

80%. 90%   100-200%

fun?

→ 50%   80%.

$\boxed{10^8\%}$