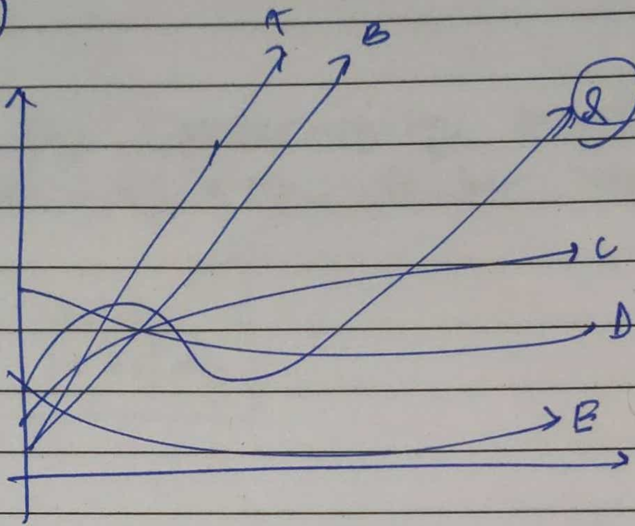
Time Complexity Analysis.

1 →
2 →

3 → Any Good way
4 → now inf
5 → Time Analysis
6 → Testing



For 8 = ? Best E
worst B

- Best Case → highest UB
  Worst Case lowest UB

Big(O) if $f(n) = O(g(n))$
then $f(n) \leq c \cdot g(n)$

(Rules - ~~Box~~

TC

Amortized (P2P) — Runtime (Online Ass)

- Somebody there is to hear you out.

Q. 100 n log n

$100 \times 10^5 \times \log_{10} 10^5$

500×10^5 Ans

$10^8 \times 5$ Ans

- Drop Constant
- Do not drop constant

AP, GP, HP.

a
a+d
a+2d
⋮
a+(n-1)d

$T_n = a+(n-1)d$

$S_n = \frac{n}{2}\left[2a + (n-1)d\right]$

$= \frac{n}{2}\left[a + l\right]$

$\frac{4-1}{2}$

GP.

a, ar, ar^2, ... ar^{n-1}

$S_n = \frac{a(1-r^n)}{1-r}$ $\forall |r|<1$

$S_\infty = \frac{a}{(1-r)}$

$\boxed{r: \text{Common Ratio}}$

HP. : reciprocal of AP.

36

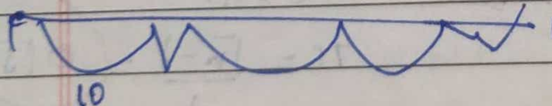
6+6

1
7
13
19
25
31

① for(i=1; i≤N; ~~i++~~) i=i+√n
cout

TC
$\frac{(\text{end}-\text{start})}{\text{Iter}}$

$\Rightarrow \frac{N}{\sqrt{n}} = \sqrt{n}$



1, 10, 100

$\frac{100}{10} = 10$

② for(i=1; i<N; i++)
for(j=1; j<N; j+=√n)
print(x);

O(n√n)

√1+√2 —— √n

$1+2+3+\dots k$

$\frac{1}{3}$

1+2 1+2+3 1+2+3+4

--

k

③ $s=1 \to 3 \to 6 \to 10$

$i=1$ 2 3 4

$i=1$ $i=2$ $i=3$ $i=4$ $i=5$

$s=2$ $s=4$ $s=7$ $s=8$ $s=12$

$s=6$ 7 8

$s=17$ 23

$n=25$

Stop when k^{th} jump $= N$

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

$$O(k^2+k) = O(2n)$$

$$O(k^2) = O(n) \Rightarrow O(k) = O(\sqrt{n})$$

```
i=1; s=1
while (s<=N)
{  i++;
   s = s+i;
   cout(x);
}
```

④ for(i=1; i^2<N; i++)

$s=1$

$e=\sqrt{N}$

$TC = \frac{\sqrt{n}-1}{1} = O(\sqrt{n})$

⑤

$O(n)$

$1+2+3 \dots N$

```
for(i=1; i<=N; i++)
   for(j=i; j<=i; ++j)
```

Replace with greater ie N

$O(n)$

$O(n)$

```
for(k=1; k<=10; ++k)
```

$O(n^2)$

Replace with the max i

$\Rightarrow i^2 = N^2$

⑥ for(i=1; i<=N; i++)
    for(j=1; j<=i^2; j++)   ~~O(n)~~ $O(n^2)$
        for(k=1; k<=N/2; k++)   O(n/2)

$\Theta(n^4)$

---

⑦ for(i=1; i<=N; i=i*2)
        print(x)

log n

j: 1 2 4 8 16 ... 2^{k-1}

i: 1 2 3

$2^{k-1} > n$

$\log_2 2^{k-1} > \log n$

$K < \log_c n + 1$

TC $\log_c n$

$\log_{10} N$ $\log_{20} N$

higher Compression factor
(how fast N can be compressed to 1)

---

⑧
O(n/2) for(i=N/2; i<=N; i++)
O(n/2)   (j=1; j<=N/2, j++)

$\frac{n}{2} \times \frac{n}{2} \times \log n$

O(log n)   (k=1, k<=N; k=k*2)
        print(x)

$n^2 \log n$

N → N/10 → N/100 → ... → 1

N → N/20 → N/400 → ... → 1

$\log n = \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} - - - \frac{1}{N}\right)$ 5

$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots \infty$ $(-1 \le x \le 1)$

(11) for(i=1; i<N; i++) $O(n)$

for(j=1; j≤N; j=j+i) logn

i=1	i=2	i=3	i=N
j=1, 2, 3 ---- N	j=1, 3, 5, ---N N/2	j = N/3 times	j = 1 time.

$N\left(1 + \frac{1}{2} + \frac{1}{3} + - - - \frac{1}{N}\right) = O\, n \log n$

weytog.

(12) $N = 2^{2^k}$ N×l

for(i=1; i≤N; i++)

j=2;

while (j≤N)

{ $j = j^2$

}

log log n

idiot

2^{2^k}

i=1

2, 4

$2, 2^2, 2^8, 2^{16}$

2^{2^k}

i=1

2, 4, 16, 16^2 ---- 2^{2^k}

1 2 3 4 2

2^{2^0} 2^{2^1} 2^{2^2} 2^{2^3}

$2^{2^k} > N$

$\log 2^{2^k}$ $\log N$

2^k $\log N$

$\log 2^k = \log \log N$

$k = \log \log N$

Jir Explanation

$j \to \cancel{2}$	$K = 2$	$K = 3$	$K = 4$
$\cancel{8} \to N$	$N = 2^{2^2} = 2^4$	$N = 2^{2^3} = 8$	$N = 2^{2^4} = 2^{16}$
$K = 1$	$j = 3$ times	$j = 4$ times.	$J = 5$ tin
$N = 2^8$	$2 \to 2^2 \to 2^4$		$2 \to 2^2 \to 2^4 \to$
$j = 2$ tin			$2^8 \to 2^{16}$
$2 \to 2^2$			

$\Rightarrow j = (k+1)$

$2^{2^k} = N$

$\log_2 2^{2^k} = \log_2 N$

$2^k = \log_2 N.$

$\log_2 2^k = \log_2 \log N$

$k = \log(\log N)$

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

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

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

$\vdots$

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

$= 3 + T(n-3)$

$= \cancel{k}$

$T(n-1)$

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

$= 2 + T(n-2)$

$\vdots$

$k + T(n-k)$

$\vdots$

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

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

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

$\underline{\underline{n}}$

1

2

4

⋮

$2^n - 1$

$S_\infty = \frac{1(2^n - 1)}{2-1} =$

Masters Theorem

$T(n) = aT(n/b) + \theta(n^k \log^p n)$

$a \geq 1,\ b > 1,\ k \geq 0$ | $p \to$ real No.

O = WC

Ω = BC

θ = Avg Case

1) $(a > b^k)$ then $T(n) = \theta(N^{\log_b a})$

2) $a = b^k$ then $p > -1$, then $T(n) = \theta(N^{\log_b a} \cdot \log^{p+1} N)$

$p = -1$ then $T(n) = \theta(N^{\log_b a} \cdot \log\log N)$

$p < -1$ then $T(n) = \theta(N^{\log_b a})$

3) $(a < b^k)$ then $p \geq 0$, then $T(n) = \theta(n^k \log^p n)$

$p < 0$, then $T(n) = \theta(n^k)$

Q1 $4T(n/2) + n^2$

$a = 4 \quad b = 2 \quad k = 2 \quad p = 0$

2 i) $a = b^k$

$$\left[n^{\log_2 4} \cdot \log n\right]$$

$\Rightarrow (n^2 \log n)$

Q2

$T(n) = 2T(n/2) + \sqrt{n}$

$a = 2 \quad b = 2 \quad k = \frac{1}{2}$

$b^k = 2^{1/2} = 1.4$

$a > b^k \quad \theta\left(n^{\log_2 a}\right) = N^1 = n$

Leetcode

$\Rightarrow$ $< 10^8$ Computations $\rightarrow$ 1sec. 30% acceptance

$1 \leq N < 20$ 2^n will work

$1 < N < 10^4 \Rightarrow O(n \log N), n\sqrt{n}$

$10^8 \rightarrow 1$sec $O(n)$

$10^{12} \Rightarrow \log n$ $10^6 \rightarrow 1$M

$10^9 \Rightarrow \log n, \sqrt{n}$ $10^9 \rightarrow 1$B

$\log_2 10^9 \Rightarrow 32$

$10^7, 10^6 \Rightarrow n$

$\log_2 10^8 : 28$

$10^5 \Rightarrow n \log n$

$10^4 \Rightarrow n \log n, n\sqrt{n}$

$10^3 \Rightarrow O(n^2)$

$10^2 \Rightarrow n^3$

$1 < N < 10^5$

- Recursion ✗ 2^n
- DP $O(n^2)$ ✗ if 10^3 then ✓
- Greedy / BS ✓