Time Limit Exceeded ✗

Hardware dependent

Speed 2 GHz ??

2×10^9 Hz

2×10^9 operations in 1 s

1 s = 2×10^9

2 s = 4×10^9 operations

Bubble sort ✗ ✓

$n \Rightarrow n \cdot 10^5$

$O(n^2)$

2 s

10^{10} = 4×10^9

10×10^9 X = 4×10^9

100

50000 ✓

90000

2.5×10^9 ✓

8.1×10^9 ✗

10^9

$T(N) = K \cdot 10^9 + C$

10^8 → operations ✓ ✓

Sabsi jagah ✓

B.S

$\log n$

$n = 10^5$

$n = 10^{10}$ ✓

$n = 10^{20}$ ✓

$n = 10^{10^9}$ ✗

$10 \times K$

$n = 10^5$

$n \cdot \log n \rightarrow 10^5 \times 5 \times C$

$n = 10^7$ ✓ ✓

$n = 10^8$??

$8 \times 10^8 \times C$

$12 \times 10^8 = 1.2 \times 10^9$ ✗

L.S $O(n)$

$n = 10^6$ ✓

$10^9 = ??$

$n = 10^8$ ✓

$n! = n \times (n-1) \times (n-2) \times (n-3) \ldots 3.2.1.$

n : # permutations

Example

n<=10	O(n!) O(2^n)	permutation Subsets
n<=100	O(n^4)	4 nested Loop
n<=400	O(n^3) ✓	Floyd warshell Graph
n<=2000	O(n2logn)	2 nested loops + BS
n<=10^4	O(n2)	Bubble.Selection.Insertion
n<=10^6	O(nlogn)	Merge, Quick ✓, Aggressive Loops
n<=10^8	O(n)	LS
n<=10^18	O(log(n))	BS ✓

long 10^9 1.5 ✓

$2000 \times 2000 \times \log(2 \times 10^3)$

$20 \times 10^6 \times 5 = 2 \times 10^7$ ✓

5040

$3.6 \times 10^6 \leq 10^8$ ✓

$11! = 4 \times 10^7$ ✓

$12! =$

agr =

```
for (i) {→n
    for(j) {→n
        for(k) {→n
            for(l) {→n
            }
        }
    }
}
```

$1 \leq n \leq 100$

n^4 ✓

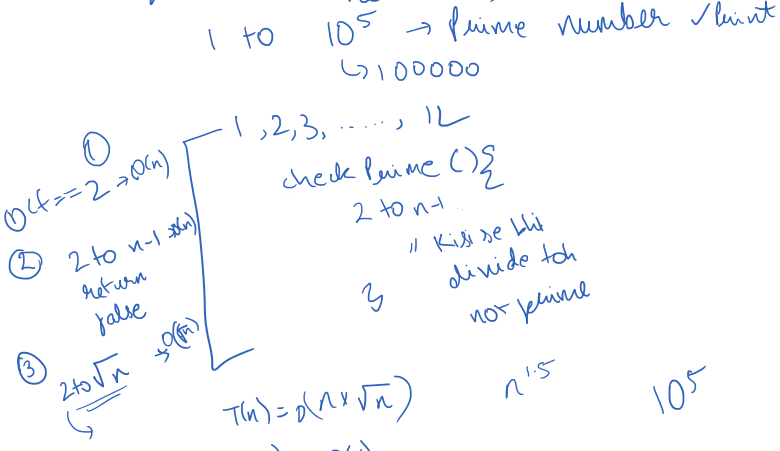
$(10^2)^4 \approx 10^8$ ✓

Sieve of Eratosthenes (SOE)

24 December 2022 10:25

1 to 10^5 → prime number / limit

↳ 100000

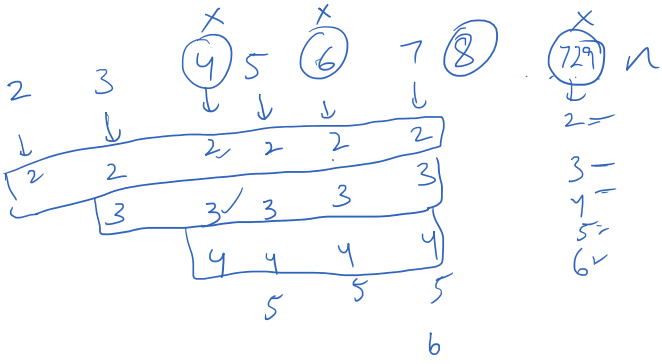


$T(n) = O(n \times \sqrt{n})$

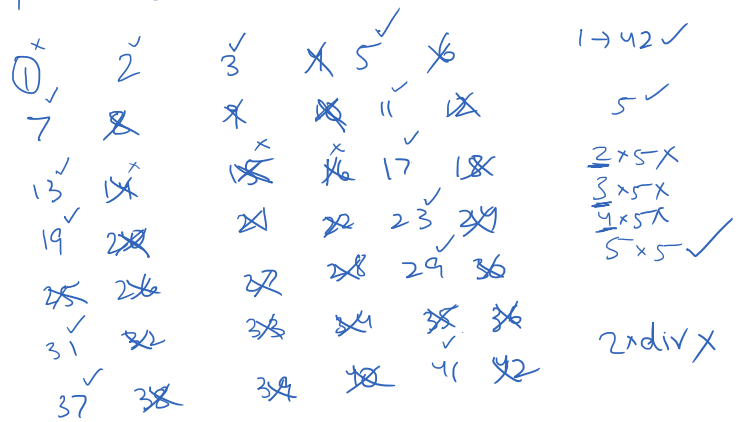
$S(n) = O(1)$

$n^{1.5}$ 10^5 Chalak

div



1 → 50



1 → 42 ✓

5 ✓

2×5 ✗

3×5 ✗

4×5 ✗

5×5 ✓

2, 11, 13, 17, 19, 23, 29, 31, 37, 41 ✓

2 × div ✗

```
for (int div = 2; div <= n; div++) {
    if (isPrime[div] == true) {
        for (int kata = div * div; kata <= n; kata += div) {
            isPrime[kata] = false;
        }
    }
}
```

for (i = 4; i ≤ n; i += 2) { } → $\frac{n}{2}$

for (i = 9; i ≤ n; i += 3) { } → $\frac{n}{3}$

$T(n) = \frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7} + \frac{n}{11} + \dots$

↳ $n \times \log(\log n)) \longrightarrow n$

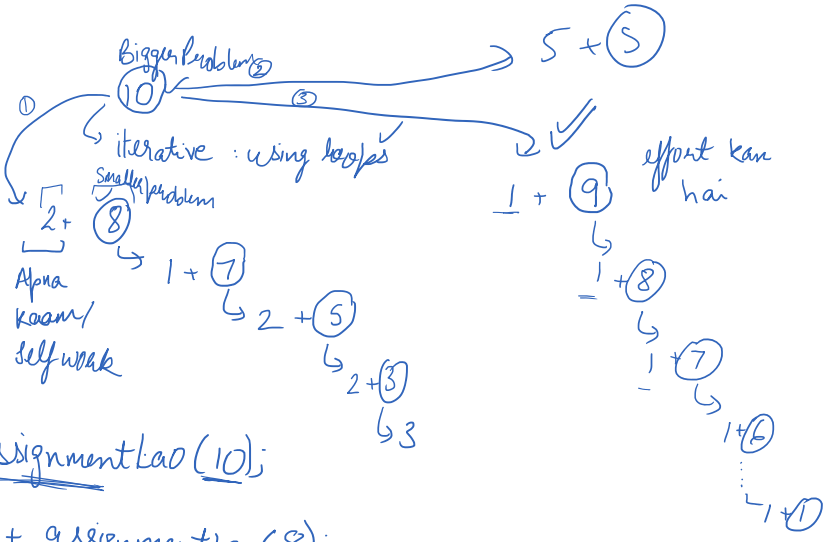
$S(n) = O(n)$

14 December 2022 11:33

Recursion → It is a problem solving technique.

A function which calls itself.

↳ the task is to break any problem into smaller problems, assume that the smaller problem will work and do the corresponding self work.



assignmentLao(10);

↳ 2 + assignmentLao(8);

Steps to be followed for solving recursion:

1) Identify the bigger problem, smaller problems

we can have multiple smaller problems

2) From all the smaller problems, choose the one which requires the least effort ⟶ the corresponding self work should be min.

3) Assume that the smaller problem works fine

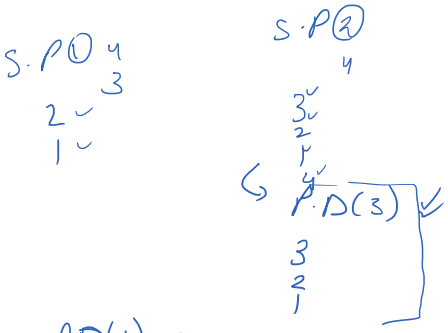
4) Do the self work and create your solution.

5) # Base Case → Iski smaller problem nahi hai

Question (1) Print Decreasing, n

P.D(n)

Ex: PD(4) : 4 → 3 2 1 PD(3) P.D



PD(1) → self work S.P

1

PD(0)

0

PD(-1)

-1

PD(-2)

-2

(2) P.(n) :

② PI(n):

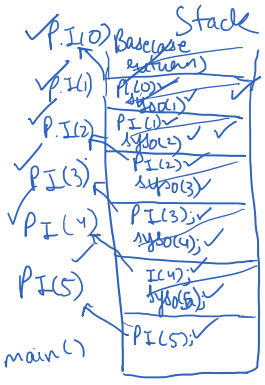
Ex PI(5) → 1 2 3 4 5

B.P ⇒ PI(5)

↳ PI(4)

1 2 3 4

+5 → self work



→ func^{n} frame
↳ local variable

1 2 3 4 5

Heap
↳ new vale
↳ non primitive

① 4.W P.D.I(n)

3 ← PDI(3)
2
1
1
2
3

② PID(n)

1 PID(3)
2
3
3
2
1