Todays Content

1. Steps followed to solve a question
2. Importance of contraints
3. Idea/datatype using constraints
4. Space Complexity

Few Maths:

$a^m * a^n = a^{m+n}$

$\frac{a^m}{a^n} = a^{m-n}$

$(a^m)^n = (a^{m \times n})$

int range = $\{-2*10^9 \ldots 2*10^9\}$

long range = $\{-8*10^{18} \ldots 8*10^{18}\}$

Given Questim:

1. Problem statement
2. Input format
3. Output format
4. Constraints
5. Sample Test Cases
6. Explanation of Testcases.

Steps followed to solve questim.

1. Read Questim → Idea → Correctness → Code → TLE > 10^8

Idea → Correctness → Code → TLE > 10^8

Idea → Correctness → Code → Submited.

Q: Given an ar[N] check if pair (i,j) exists such that their sum = k

Constraints:

$1 <= N <= 10^6$

$1 <= ar[i] <= 10^9$

Given ar[N] & k

```
for(int i=0; i<N; i++){
    for(int j=0; j<N; j++){
        if(ar[i]+ar[j]==k){
            return true;
        }
    }
}
return false;
```

$1 <= N <= 10^6$

TC: $O(N^2)$: TLE.

↳ Max $N=10^6 \Rightarrow N^2 = 10^{12} > 10^8$ iterations

Questim → Idea → CorrectNot → TC → Wrk/TLE → Code → Submit

Pratice

Idea based on constraints

$2^{10} = 1024 \approx 1000 = 10^3$

$\log_2 10^3 \approx \log_2 2^{10} = 10$

$\log_2 AB = \log_2 A + \log_2 B$

$\log_2 10^6 = \log_2 10^3 * 10^3 = \log_2 10^3 + \log_2 10^3 = 20$

Ex1: $1 \le N \le 10^3$

$1 \le arr[i] \le 10^4$

Accepted TC:

Idea1: $O(N^3) = (10^3)^3 = 10^9 > 10^8$ TLE

Idea2: $O(N^2) = (10^3)^2 = 10^6$ ✓

Idea3: $O(N \log N) = 10^3 \log_2 10^3 = 10^3 * 10 = 10^4$ ✓

Idea4: $O(N) = 10^3$ ✓

Ex2: $1 \le N \le 10^6$

$1 \le arr[i] \le 10^6$

Accepted TC:

Idea1: $O(N^2) = (10^6)^2 = 10^{12} > 10^8$ TLE

Idea2: $O(N \log N) = 10^6 \log_2 10^6 = 10^6 * 20 = 2 * 10^7$ ✓

Ex3: $1 \le N \le 25$

$1 \le arr[i] \le 10^6$

Accepted TC:

Idea1: $O(N!) = 25! > 10^8$ TLE

Idea2: $O(2^N) = 2^{25} = 10^3 * 10^3 * 32 = 3.2 * 10^7$ ✓

↳ Subsets.

Ex4: $1 \le N \le 10$

$1 \le arr[i] \le 10^5$

Accepted TC:

Idea1: $O(N!) = 10! = 3.6 * 10^6$ ✓

↳ Permutation

Datatype based on constraints

Q1: Given ar[] calculate sum of array elements

Constraints:

$1 <= N <= 10^5$

$1 <= ar[i] <= 10^9$

Sum range : $\{1 .. 10^{14}\}$ >> Int range

Min : $ar[1] = \{1\} = 1$

Max : $ar[10^5] = \{10^9 + 10^9 + 10^9 + \ldots 10^9\} = 10^5 * 10^9 = 10^{14}$

```
int sum=0;
for(int i=0; i< N; i++){
    sum = sum + ar[i];
}
print(sum);
```

$10^9 + 10^9 \rightarrow 2 \cdot 10^9$

$10^9 + 10^9 + 10^9 \rightarrow 3 \cdot 10^9$

Note1: Constraints will also tell about range of variables, based on that we can estimate datatype of variable.

Q: For below constraints, range of sum variable

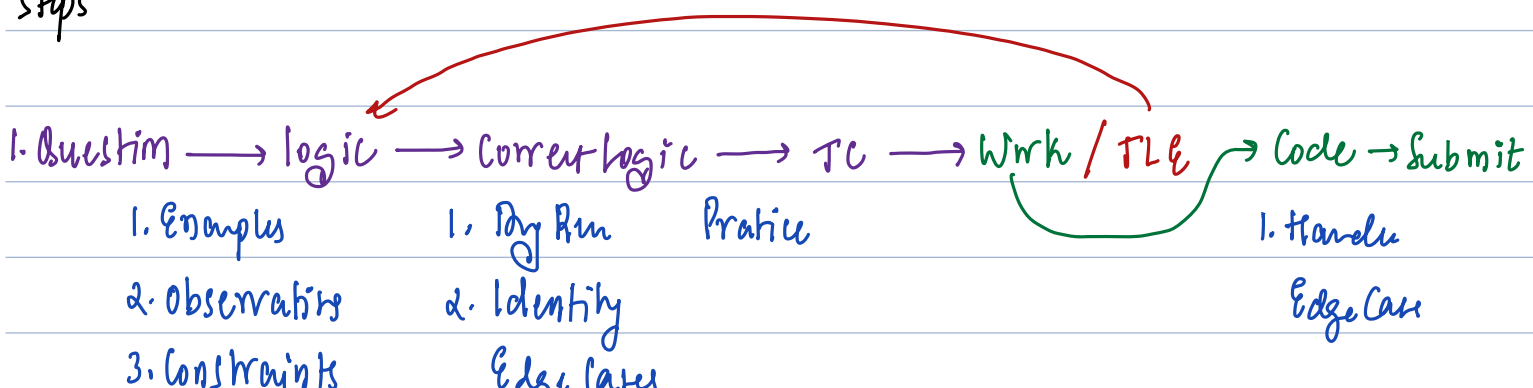
$1 <= N <= 10^5$

$-10^9 <= ar[i] <= 10^9$

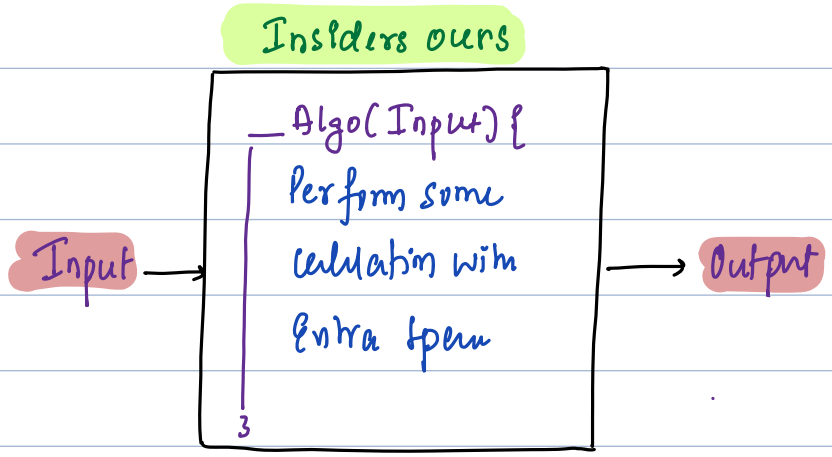
Min: $ar[10^5] = \{-10^9 \ -10^9 \ -10^9 \ldots -10^9\} = -10^9 * 10^5 = -10^{14}$

Max: $ar[10^5] = \{10^9 \ 10^9 \ 10^9 \ldots 10^9\} = 10^9 * 10^5 = 10^{14}$

Steps



Space Complexity: Max extra space used by algorithm during it's execution.



Note1: We neglect Input & Output space, & only consider extra space taken by Algo

Note2: We use BigO to analyze space complexity as well

↳ Input

```
void Algo1(int N){          int = 4B   long = 8B
    int a = N;      → 4B
    int y = n*n     → 4B    Total space = 16B → Constant space = O(1)
    long z = n+y    → 8B
}
```

```
void fun(int N){
    int arr[10];    → 40B
    int n, y;       → 8B    Total Space = 56 + 4N → O(N)
    long z;         → 8B
    int a[N];       → 4N
}
```

```
void func(int N){
    int n = N;          ⟶ 4B
    int y = n*n;        ⟶ 4B
    long z = n+y;       ⟶ 8B
    int arr[N];         ⟶ 4N
    long d[N][N];       ⟶ 8N^2
}
```

↳ matrix of size = N*N

Total space = 16B + 4N + 8N^2 = O(N^2)

Q: Given an arr[N] return max of an array.

Input

```
int maxarr(int arr[], int N){
    int max = INT_MIN;
    for(int i = 0; i < N; i++){
        if(arr[i] > max){
            max = arr[i];
        }
    }
    return max;
}
```

Iteration = N ⟶ O(N)

Total space = 4B ⟶ O(1)