

Today's Content

1.

2.

3. Tricky Interview questions

Few Maths:

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

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

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

$$\text{int range} = \{-2 \times 10^9 \dots 2 \times 10^9\}$$

$$\text{long range} = \{-8 \times 10^{18} \dots 8 \times 10^{18}\}$$

Given Question:

1. Problem statement

2. Input format

3. Output format

4. Constraints

5. Examples

6. Explanation

Steps followed to solve question.

1. Question → Idea 1 → Correctness of logic → Code → TLE $> 10^8$

Idea 1 → Correctness of logic → Code → TLE $> 10^8$

Idea 2 → Correctness of logic → Code → Submit

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

Constraints:

$$1 \leq N \leq 10^6$$

$$1 \leq arr[i] \leq 10^9$$

Idea: Generate all pairs.

for (int i=0; i<N; i++) { Iterating = $N + N^2 = O(N^2)$: TLE.

 for (int j=0; j<N; j++) {

 if (arr[i] + arr[j] == k) {

 3 3 ==

$$N = 10^6$$

$$N^2 = (10^6)^2 = 10^{12} >> 10^8$$

Question → Idea 1 → Correctness of logic → TC → Work/TLE → Code → Submit

Idea based on Constraints

→ Input / Array Size

$$\text{En1: } 1 \alpha = N \alpha = 10^3$$

$$1 \alpha = \text{arbitrary} \alpha = 10^4$$

Accepted TC:

1. TC: $\Theta(N^3) = (10^3)^3 = 10^9 > 10^8 \text{ TLE}$
2. TC: $\Theta(N^2) = (10^3)^2 = 10^6 \approx 10^8 \checkmark$
3. TC: $\Theta(N \log N)$ ✓
4. TC: $\Theta(N)$ ✓

$$\text{En2: } 1 \alpha = N \alpha = 10^6$$

$$1 \alpha = \text{arbitrary} \alpha = 10^6$$

$$\log_{\frac{1}{2}} 10^6 = 20$$

Accepted TC:

1. TC: $\Theta(N^2) = (10^6)^2 = 10^{12} \text{ TLE}$
2. TC: $\Theta(N \log N) = 10^6 * 20 = 2 * 10^7 \checkmark$
3. TC: $\Theta(N)$ ✓

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

$$2^{20} \approx 10^6$$

$$2^{25} \approx 2^5 * 10^6 \\ = 32 * 10^6$$

$$\text{En3: } 1 \alpha = N \alpha = 25$$

$$1 \alpha = \text{arbitrary} \alpha = 10^6$$

Accepted TC:

1. TC: $\Theta(N!) = 25! \gg 10^8 \text{ TLE}$
 2. TC: $\Theta(2^N) = 2^{25} = 3 * 2 * 10^7 \approx 10^8 \checkmark$
- ↳ Estimate subsets / subsequences

$$\text{En4: } 1 \alpha = N \alpha = 10$$

$$1 \alpha = \text{arbitrary} \alpha = 10^5$$

Accepted TC: → Permutations

1. TC: $\Theta(N!) = 10! = 3 * 5 * 10^6 \approx 10^8 \checkmark$

Datatype based on constraints

Q1: Given arr[] calculate sum of array elements

Constraints:

$$\left. \begin{array}{l} 1 \leq N \leq 10^5 \\ 1 \leq arr[i] \leq 10^9 \end{array} \right\} \text{According to constraints, Range of sum.}$$

Ex: $1 \leq \text{sum } s \leq 10^{14}$ Ex: $arr[1] = \{1\}$ $arr[10^5] = \{10^9, 10^9, \dots, 10^9\}$

~~long int~~ sum=0; // Issues: sum exceeding int range. \Rightarrow long.

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

Q2: For below constraints, range of sum variable

$$1 \leq N \leq 10^5$$

$$-10^9 \leq arr[i] \leq 10^9$$

$$\text{Ex: Min: } -10^9 * 10^5 = -10^{14}$$

$$\text{Ex: Max: } 10^9 * 10^5 = 10^{14}$$

$$arr[10^5] = \{-10^9, -10^9, -10^9, \dots\}$$

$$arr[10^5] = \{10^9, 10^9, 10^9, \dots, 10^9\}$$

Approach:

Question \rightarrow Idea \rightarrow Correctness \rightarrow TC \rightarrow Work/TLE \rightarrow Code \rightarrow Submit

a. Observing

a. Dry Run

pratik

a. Datatype

b. Constraints

b. Identify

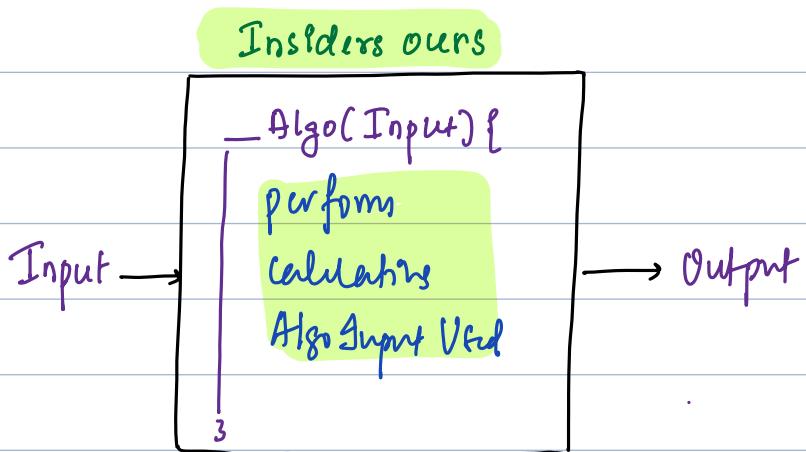
b. Handle

c. Encryptions

Edge Case

Edge Case

Space Complexity: How amount of extra space taken by Algo, during its execution



Note: While calculating space complexity neglect Input & Output space.

Consider space taken by algorithms

Note: To analyse space complexity, we will use Big O

Eg: ↳ Input $\text{int} = 4B$ $\text{long} = 8B$

void Algo1(int N){
 int x = N; $\rightarrow 4B$
 int y = $n \times n$ $\rightarrow 4B$
 long z = $x+y$ $\rightarrow 8B$
}

Total = $16B \rightarrow$ Constant space $\rightarrow O(1)$

void fun(int N){
 int arr[10]; $\rightarrow 40B$
 int x, y; $\rightarrow 8B$
 long z; $\rightarrow 8B$
 int a[N]; $\rightarrow 4N$
}

Total Space = $56 + 4N$
 $\text{Big O} = O(N)$

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]; → 8N²

Total Spac = 16 + 4N + 8N²

Big O = O(N²)

}

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

Inputs

int maxarr(int arr[], int N) { Total Spac = 4B → O(1)

Output → int max = INT_MIN; Time Comp = O(N)

for(int i = 0; i < N; i++) {
 if(arr[i] > max) {
 max = arr[i];
 }
}

return max;

}