



Quick Notes

Things to Remember

- 1 byte = 8 bits
- In general machines can perform 10^8 operations per second, so we can write our code accordingly to avoid TLE error.
- The range where our answer lies is called search space.

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    return 0;
}
```

Arrays

- If array is of allocated with fewer values than its size then the rest of the locations are by default initialized with zero

- In C++, to set them all to -1, you can use something like `std::fill_n` (from `<algorithm>`): `std::fill_n(array, 100, -1);`
- arrays are always passed by reference
- XOR is useful because of four key properties: XOR has an identity element. XOR is self-inverting. XOR is associative.

Sorting

- Selection sort has worst and best case complexity as $O(n^2)$
- Selection sort is good for small values
- A sorting algorithm is said to be stable if two objects with equal or same keys appear in the same order in sorted output as they appear in the input array to be sorted.
- Selection sort is not stable
- Bubble sort best case complexity is $O(n)$ and worst case is $O(n^2)$
- Bubble sort use case is that it gives nth largest element in nth round
- An in-place algorithm is an algorithm that does not need an extra space and produces an output in the same memory that contains the data by transforming the input 'in-place'. However, a small constant extra space used for variables is allowed.
- Bubble sort is in-place
- Insertion Sort is adaptive, that means it reduces its total number of steps if given a partially sorted list, hence it increases its efficiency. Its space complexity is less. Insertion sort requires a single additional memory space.
- Insertion sort best case complexity is $O(n)$ and worst case is $O(n^2)$