BRACU CP Workshop

Day 7

Arman Ferdous, Shafin Alam

BRAC University, Dhaka

7 May, 2025

Most important things to reach Cyan/Blue

Code Faster

- div2A $<$ 5min, div2B $<$ 20min, ...
- 2090A - There is a treasure at a.5 meters underground. Everyday Alice & Bob take turns to dig out the chest, starting with Alice.
- Alice will always dig x meters, Bob will always dig y meters. Who will find the treasure?
- $a, x, y \leq 10^9$.
- If you can solve quickly, you give yourself more time for the harder problem!

Most important things to reach Cyan/Blue

Topics to Master

- Math, Dynamic Programming, Greedy, Binary Search, Bruteforce
- Graph Theory, Data Structures, Probability & Expectations
- A training guide by a red coder
- Practice by tag in codeforces for topic improvement
  Practice at atcoder to improve speed.

Wonderful Gloves - 2096B

- There are some colorful gloves, the ith color has l_i left glove and r_i right glove.
- You want to find k matching pairs from them, with different colors, in the dark.
- Find the minimum number of gloves to pick to ensure this.
- Example 1:
  1 1
  100
  1
  Answer $=$ 101
- Example 2:
  3 2
  100 1 1
  200 1 1
  Answer $=$ 303

Wonderful Gloves - 2096B

- Maximum number of gloves we must pick so that no matching pair of gloves exist?

Wonderful Gloves - 2096B

- Maximum number of gloves we must pick so that no matching pair of gloves exist?
- What is the maximum after that that we do not get k different colors?
- It is enough to consider maximum as this is the worst situation that can happen. And if we do not pick this much we always run the risk.

Invariants & Monovariants

- Invariant is some property that does not change throughout some operations.
- Suppose, you have an array A. In one operation you can pick any two adjacent indices and set $a_i := a_i + 1$ and $a_{i+1} = a_{i+1} - 1$. What never changes?
- The sum of all elements $\sum a_i$ stays constant.
- Now if you are given another array B with same length, and asked to check if you can transform A into B, then the invariant is a necessary condition to satisfy.

Invariants & Monovariants

- Monovariant is similar, but now instead of staying constant it changes in single direction (only increases or decreases).
- Suppose, in an array you perform the following whenever possible:
  if $a_i > a_{i+1}$ then swap a_i and a_{i+1}.
  Is there any monovariant here?
- The number of inversions in the array will always decrease by 1.
- If any array is unsorted, then there exists at least one index i such that $a_i > a_{i+1}$. The swap will decrease it by 1. Once it touches 0 the array will get sorted.
  Do you know this sorting algorithm?

- Useful blog 1, Useful blog 2.
- Given n integers $a_1, ..., a_n$ you can do the following operation:
  Select an i such that $2 \leq i \leq n - 1$.
  Set $a_i := a_{i+1} + a_{i-1} - a_i$.
- Report whether you can transform into another array $b_1, ..., b_n$ in any number of operations. (1110E)
- Also try this problem out: (2084C)

Wonderful Lightbulbs - 2096D

Bit easier example

- The 2D plane has bulbs at each point, all turned OFF. A secret location has a treasure and only that bulb is ON.
- To hide its location, several updates were made. In each update a (x, y) is chosen, and then the bulbs at $(x, y), (x, y + 1), (x + 1, y - 1), (x + 1, y)$ are flipped.
- Finally n light bulbs are ON, and you are given the list of them. Print where the treasure was.

Wonderful Lightbulbs - 2096D

Solution

Interactive Problems

- Problems where you interact with the input real time.
- 727C A hidden array of length $3 \leq n \leq 5000$ exists. You can ask for the sum of any two indices i, j where $i \neq j, 1 \leq i, j \leq n$.
- Find the whole array in at most n queries. It is guaranteed that each a_i is a positive integer not greater than 10^5.

Interactive Problems

727C Solution

Interactive Problems

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

  int query (int i, int j) {
      cout << "? " << i << " " << j << endl;
      int sum; cin >> sum;
      return sum;
  }

  int main() {
      ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
      int n; cin >> n;
      int P = query(1, 2), Q = query(2, 3), R = query(1, 3);

      vector<int> a(n + 1);
      a[1] = ..., a[2] = ..., a[3] = ...;
      // solve for a[4], ... a[n]

      cout << "!";
      for (int i = 1; i <= n; ++i) {
          cout << " " << a[i];
      } cout << endl;
  }
  ```
  ```

Object Identification - 2067D

- The judge has a hidden object, it is either a directed graph with n nodes, or the 2D plane with n points.
- If it is the digraph, then there are n edges, $x_i \to y_i$.
- If it is the 2D plane, then there are n points, (x_i, y_i).
- You only know $x_1, \ldots, x_n$, but the y array is hidden. You only need to know which type of object judge has. You can query the following:
- Select two integers i, j where $1 \leq i, j \leq n$ and $i \neq j$. You recieve a number:
  If graph, then the shortest distance from vertex i to vertex j (0 if unreachable).
  If plane, then $|x_i - x_j| + |y_i - y_j|$.
- It is guaranteed that (x_i, y_i) pairs are all distinct and $x_i \neq y_i$. Use at most 2 queries to figure it out.

Chimpanzini Bananini - 2094G

- Score of an array b with m elements $= \sum_{i=1}^{m} b_i \times i$.
- Given an initially empty array, you need to report its score after each update. There will be $q \leq 2 \times 10^5$ updates.
- append x: $[a_1, ..., a_n] \to [a_1, ..., a_n, x]$
- cyclic shift: $[a_1, a_2, ..., a_n] \to [a_n, a_1, a_2, ..., a_{n-1}]$
- reverse: $[a_1, ..., a_n] \to [a_n, ..., a_1]$

Game on a Palindrome

- Alice and Bob play a game on a binary string (only 0 and 1) that starts as a palindrome.
- They take turns, starting with Alice.
- On each turn, a player can either:
  - Change a 0 to 1 by paying 1 dollar, or
  - Reverse the whole string for free, only if it's not a palindrome and the last move wasn't a reverse.
- The game ends when the whole string becomes all 1s.
- The player who spends less money wins. If both spend the same, it's a draw.

Constraints

- $1 \leq n \leq 10^3$ (length of the palindrome string)
- s is a palindrome of length n, consisting only of characters '0' and '1'.
- s contains at least one '0'.

Examples

1. **Even number of zeros.**

$$n = 8, \quad s = 10011001$$

2. **Odd number of zeros.**

$$n = 5, \quad s = 01010$$

Observations

- The string is a palindrome from the beginning.
- If there is a `'0'` at position i, then there is also a `'0'` at position $n - i + 1$.
- Zeros always come in matching pairs from both ends.
- If the string length is odd, the middle character can be a `'0'` without a pair.

Case 1: Exactly 0 '0's

- The string already has all '1's.
- No moves are needed.
- Both players spend 0 dollars.
- Result: **Draw**.

Case 2: Exactly 1 '0'

- Alice goes first and must flip the only '0'.
- She spends 1 dollar.
- Bob plays no move and spends 0 dollars.
- Result: **Bob wins**.

Case 3: Even Number of '0's

- Bob mirrors Alice's moves to keep the string a palindrome.
- The game reaches a point where only two zeros are left.
- Alice flips one, then Bob reverses the string.
- Alice is forced to flip the last zero.
- **Bob wins**.

Case 4: Odd Number of '0's (more than one)

- Alice flips the middle zero to make the count even.
- The game follows the same pattern as Case 3.
- Bob ends up in the same trap — flips one, Alice reverses.
- Bob is forced to flip the last zero.
- **Alice wins**.

Problem: Maximize XOR Contribution

Given an array of n integers: $a_1, a_2, \ldots, a_n$.

Pick any element a_k from the array.
Now compute the sum of XORs between a_k and every element:

$$(a_k \oplus a_1) + (a_k \oplus a_2) + \cdots + (a_k \oplus a_n)$$

Your goal is to find the maximum value of this sum over all valid choices of a_k.

Constraints:

- $1 \leq n \leq 2 \cdot 10^5$
- $0 \leq a_i < 2^{30}$

Example

Let the array be:

$$a = [1,\ 3,\ 4,\ 6,\ 9]$$

We choose $k = 5$, i.e., $a_k = 9$, and compute:

$$(9\oplus 1)+(9\oplus 3)+(9\oplus 4)+(9\oplus 6)+(9\oplus 9) = 8+10+13+15+0 = 46$$

Answer: 46

Problem: Flipping Bulbs

You have n light bulbs, all initially ON.

For each $i = 1$ to n, flip the state of every bulb at position j that is divisible by i.

After all operations, some bulbs will still be ON.

Find the smallest n such that exactly k bulbs remain ON in the end.

Constraints:

- $1 \leq k \leq 10^{18}$

Problem: Fortune Telling

Alice and Bob are each given the same array of non-negative integers, $a[1 \dots n]$.

Alice starts with $d = x$; Bob starts with $d = x + 3$.

For each $i = 1$ to n, each player independently updates their d by choosing one of:

$$d \leftarrow d + a[i] \quad \text{or} \quad d \leftarrow d \oplus a[i].$$

After all n steps, Alice ends with y_1 and Bob ends with y_2.

It is guaranteed that exactly one of them can reach the given final value y.

Determine who reached the final number y: Alice or Bob.

Constraints

- $1 \leq n \leq 10^5$
- $0 \leq x \leq 10^9$
- $0 \leq y \leq 10^{15}$

Observation: Parity Matters

- Alice starts with $d = x$ (Even/Odd), Bob starts with $d = x + 3$ (Opposite Parity of Alice).
- Addition: Parity changes if added number is odd, stays same if even.
- XOR: Parity changes if XOR-ed with odd, stays same if even.
- Alice and Bob's results follow this pattern:

Solution Approach

- Final Value:
  - If the sum of the array is **even** $\rightarrow$ Parity stays the same.
  - If the sum is **odd** $\rightarrow$ Parity flips.
- Determine who can reach y:
  - Check y's parity:
    - If y matches Alice's expected parity $\rightarrow$ Alice can reach it.
    - If y matches Bob's expected parity $\rightarrow$ Bob can reach it.

Problem: Rearranging a Zero-Sum Array

You are given an array a such that the total sum is zero:

$$a_1 + a_2 + \cdots + a_n = 0$$

Your goal is to rearrange the elements so that the maximum absolute subarray sum is strictly less than the difference between the maximum and minimum elements:

$$\max_{1 \le l \le r \le n} \left| \sum_{i=l}^{r} a_i \right| < \max(a) - \min(a)$$

Determine if such a rearrangement exists.
If it does, print any valid arrangement.

Constraints

Constraints:

- $1 \leq n \leq 300,000$
- $-10^9 \leq a_i \leq 10^9$

Example:

$n = 7$

`arr = [-18, 13, -18, -17, 12, 15, 13]`

Answer:

`[15, -17, 13, -18, 13, -18, 12]`

Good Bye!