T181 Pig’s CD

Subtask 1

- For each query, do a nested for loop find l, r such that gcd(a[l].. a[r]) = x
- Time complexity = $O(Q * N^2 * \log(1000))$

Subtask 2

- First precompute an answer array ANS. Which ANS[x] is the answer for query(x)
- Obviously, a nested for loop can do that
- $O(N^2 * \log(10^6))$

Subtask 3

- $S[i]$ only equal 1 or 2
- For each i such that $S[i] = 1$, add $i * (n - i + 1)$ in $ANS[1]$
- For each i such that $S[i] = 2$, find the closest $[l, r]$ such that $a[l..r] = 2$, add $(i - l + 1) * (r - i + 1)$ in $ANS[2]$, $i * (n - i + 1) - (i - l + 1) * (r - i + 1)$ in $ANS[1]$
- $O(N)$

Subtask 4

- Notice that if we enumerate i as the starting point of the interval, the gcd for [i, j] will at most change log(1000) times.
- Now enumerate the left interval point, do a binary search for log(1000) times to find all the changing points.
- Getting the interval gcd could be done using segment tree
- $O(N * \log^2(1000) * \log^2(N))$

Subtask 5

- For speed up the process of getting the interval gcd, one could use sparse table instead
- $O(N * \log^2(10^6) * \log(N))$

Subtask 6

- The are multiple ways to solve this subtask based on subtask 5
- The easiest one is don't find the gcd but find if the two interval is divisible by gcd in the sparse table
- -> leftIntervalGCD % gcd == 0 && rightIntervalGCD % gcd == 0
- If this is still not fast enough, one could add leftIntervalGCD >= gcd && rightIntervalGCD >= gcd for avoiding the mod operator for sometimes
- $O(N * \log(10^{18}) * \log(N))$