Harry Potter - Editorial

Difficulty:

Hard

Prerequisites:

Segment Tree - (Tutorial)
Number Theory
Fibonacci Numbers - (Info)
Matrix Exponentiation - (Tutorial)

Problem in Brief:

Given an Array A and Q queries consisting of 2 integers l and r. For each query find the value of gcd(F[l], F[l + 1], …. F[r]) where F[i] is the ith Fibonacci number and gcd is the Greatest Common Divisor Function.

Editorial:

First, let us try to find the gcd of 2 Fibonacci numbers.

We claim that gcd(F[m], F[n]) = F[gcd(m, n)]

Proof:

Harry Potter - Editorial

As you suggest, we can use that

$$f_{m+n} = f_{m-1}f_n + f_m f_{n+1}.$$

We first show that any two consecutive Fibonacci numbers are relatively prime. i.e. We show that

$$\gcd(f_{n+1}, f_n) = 1.$$

This is clearly true when $n = 1$. Suppose that it holds for some n. Then we have that (using the Euclidean Algorithm)

$$\gcd(f_{n+2}, f_{n+1}) = \gcd(f_{n+1}, f_{n+2} - f_{n+1}) = \gcd(f_{n+1}, f_n) = 1$$

where in the last two steps we used the recurrence relation for the Fibonacci numbers, and the induction hypothesis. Thus the claim holds for $n + 1$ as well, and hence for all n by induction.

We then proceed as follows:

Note that for any natural numbers m and n, that

$$\gcd(f_{m+n}, f_n) = \gcd(f_{m-1}f_n + f_m f_{n+1}, f_n) = \gcd(f_m f_{n+1}, f_n) = \gcd(f_m, f_n)$$

we the last equality follows from the fact that $\gcd(f_{n+1}, f_n) = 1$.

We thus see that calculating the greatest common divisor of two Fibonacci numbers becomes equivalent to applying the Euclidean Algorithm *to the subscripts*.

In other words, if we iteratively apply the relation

$$\gcd(f_n, f_m) = \gcd(f_m, f_{n-m})$$

for $n \geq m$, and we keep track of the indices of the two Fibonacci numbers involved as we go, we see that pairs of numbers that are the indices are identical to the pairs of numbers we would obtain if were were using the Euclidean Algorithm to calculate the greatest common divisor of m and n.

Thus the pairs of indices involved will eventually be $\gcd(m, n)$ and 0. We thus conclude that

$$\gcd(f_m, f_n) = \gcd\left(f_{\gcd(m,n)}, f_0\right) = \gcd\left(f_{\gcd(m,n)}, 0\right) = f_{\gcd(m,n)}.$$

Now to answer any query we just need to find the gcd of consecutive numbers in the array. We can do that using Segment tree as gcd is a commutative function.

There is still the problem of find Fibonacci upto 10^9.
To overcome this problem, we can use Matrix Exponentiation.

Harry Potter - Editorial

As finding Fibonacci using Matrix Exponentiation is a basic application, we can now find the answer to each query.

Time Complexity:

For each query we compute the gcd function log(N) times and gcd function can take upto O(A[i]) time, Hence the Time Complexity is

O(Q * log(N) * log(A[i]))

Similar Problems:

First
Second
Third
Fourth