

Kağan ve Ceyhun Çözüm

Soruda yapılması gereken gözlem her sorgu için cevapları sorgu geldiği an hesaplamak yerine, bütün queryleri önce işleyip, queryleri bir ağaç formatına getirdikten sonra, sorguların verildiği sırada gezmek yerine, oluşturduğumuz ağaç içinde derinlik öncelikli arama ile, sorguları hesaplamamız gerekiyor. Dikkat etmemiz gereken bir nokta ise, ağaç üzerinde her çocuk düğüme(sorguya) gittiğimizde sorguda yapılan değişikliği uygulayıp cevabı hesaplayıp, bütün dallanmalar bittikten sonra da o anki sorgunun eski halina getirmek gerektirir. Bu sayede çözümü $O(N+Q)$ zaman karmaşıklığında yapabiliriz. Ek olarak, sorunun kodlamasını kolaylaştıracak gözlemlerden bir tanesi de, bütün 10 01 00 11 leri saymak yerine, 1 lerin sayısı ve 0 ların sayısını saymak yeterlidir. Bunun sebebi yalnızca 1 lerin sayısı 0 ların sayısına eşitse cevap **EVET** olabilir.

Çözüm Kodu:

```
#include <bits/stdc++.h>
#define st first
#define nd second
#define mp make_pair
#define pb push_back
#define sol (k+k)
#define sag (k+k+1)
#define orta ((bas+son)/2)
#define coc g[x][i]
#define mod 1000000007
#define inf 1000000009
#define N 1000005
using namespace std;

typedef long long ll;
typedef pair < int , int > ii;

int n, q, say[2], ans[N];
char s[N];
vector < pair < int , ii > > g[N];

void dfs(int x){
    if(say[0] == say[1])
        ans[x] = 1;
    for(int i = 0; i < g[x].size(); i++){
        char eski = s[coc.nd.st];
        say[eski - '0']--;
        s[coc.nd.st] = '0' + coc.nd.nd;
        say[coc.nd.nd]++;
    }
}
```

```

        dfs(coc.st);

        say[coc.nd.nd]--;
        s[coc.nd.st] = eski;
        say[eski - '0']++;
    }
}

int main() {
    scanf("%d %d %s",&n ,&q, s + 1);

    if(n != strlen(s + 1))
        assert(0);

    for(int i = 1; i <= n; i++)
        say[s[i] - '0']++;

    for(int j = 1; j <= q; j++){
        int c, i, x;
        scanf("%d %d %d",&c ,&i ,&x);
        g[c].pb(mp(j, mp(i, x)));
    }
    dfs(0);
    for(int i = 1; i <= q; i++)
        printf("%s\n", ans[i]?"EVET":"HAYIR");
    return 0;
}

```