CF1100F

区间查询线性基。

```
#include <bits/stdc++.h>
#define maxn 500086
using namespace std;

int n, q, x, y;
int b[maxn][31], pos[maxn][31];//debug:数组开小调了一万年

inline void insert(int x, int y){
    int z = y;
    for(int i = 30, j = 1 << 30;j;i--, j >>= 1){
        if(x & j){
            if(!b[y][i]){
                b[y][i] = x, pos[y][i] = z;
                break;
            }else{
                if(z > pos[y][i]) swap(x, b[y][i]), swap(z, pos[y][i]);
                x ^= b[y][i];
            }
        }
    }
}

inline int query(int l, int r){
    int ans = 0;
    for(int i = 30;~i;i--){
        //printf("%d %d %d--\n", i, b[r][i], pos[r][i]);
        if(pos[r][i] >= l && (ans ^ b[r][i]) > ans) ans ^= b[r][i];
    }
    return ans;
}

int main(){
    scanf("%d", &n);
    for(int i = 1;i <= n;i++){
        //for(int j = 0;j <= 30;j++) b[i][j] = b[i - 1][j], pos[i][j] =
pos[i - 1][j];
        memcpy(b[i], b[i - 1], sizeof(b[i - 1])), memcpy(pos[i], pos[i -
1], sizeof(pos[i - 1]));
        scanf("%d", &x);
        insert(x, i);
    }
    scanf("%d", &q);
    while(q--){
        scanf("%d%d", &x, &y);
        printf("%d\n", query(x, y));
```

```
    }
}
```

CF461B

树上连通块，树形dp

```
#include <bits/stdc++.h>
#define maxn 100086
using namespace std;

const int p = 1e9 + 7;

vector<int> v[maxn];
int f[maxn][2];
int ans;

void dfs(int i){
    for(int j = 0;j < v[i].size();j++){
        int to = v[i][j];
        dfs(to);
        f[i][1] = (1ll * f[i][1] * (f[to][0] + f[to][1]) + 1ll * f[i][0] *
f[to][1]) % p;
        f[i][0] = 1ll * f[i][0] * (f[to][0] + f[to][1]) % p;
    }
}

int n, x;

int main(){
    scanf("%d", &n);
    for(int i = 2;i <= n;i++) scanf("%d", &x), v[++x].push_back(i);
    for(int i = 1;i <= n;i++) scanf("%d", &x), f[i][x] = 1;
    dfs(1);
    printf("%d", f[1][1]);
}
```

CF300D

卷积dp

```
#include <bits/stdc++.h>
#define maxn 1086
#define maxm 105
```

```
using namespace std;

const int p = 7340033;

int t;
int n, K;
int dep;
int f[maxn / 10][maxn], g[maxn / 10][maxn];

int main(){
    for(int i = 0;i < maxm;i++) f[i][0] = g[i][0] = 1;
    for(int i = 1;i < maxm;i++){
        for(int j = 1;j < maxn;j++){
            for(int k = 0;k < j;k++){
                f[i][j] += 1ll * g[i - 1][k] * g[i - 1][j - 1 - k] % p;
                if(f[i][j] >= p) f[i][j] -= p;
            }
            for(int k = 0;k <= j;k++){
                g[i][j] += 1ll * f[i][k] * f[i][j - k] % p;
                if(g[i][j] >= p) g[i][j] -= p;
            }
        }
    }
    scanf("%d", &t);
    while(t--){
        scanf("%d%d", &n, &K), dep = 0;
        while((n & 1) && n >= 3) ++dep, n >>= 1;
        printf("%d\n", f[dep][K]);
    }
}
```

CF319C

斜率优化。

```
#include <bits/stdc++.h>
#define maxn 100086
using namespace std;

typedef long long ll;

int n;
ll a[maxn], b[maxn], f[maxn];
int q[maxn], l, r;

inline double k(int i, int j){
    return 1.0 * (f[i] - f[j]) / (b[j] - b[i]);
}
```

```
int main(){
    scanf("%d", &n);
    for(int i = 1;i <= n;i++) scanf("%lld", &a[i]);
    for(int i = 1;i <= n;i++) scanf("%lld", &b[i]);
    f[1] = 0, q[0] = 1;
    for(int i = 2;i <= n;i++){
        while(l < r && k(q[l], q[l + 1]) <= a[i]) ++l;
        f[i] = f[q[l]] + a[i] * b[q[l]];
        //printf("%d %d %lld--\n", i, q[l], f[i]);
        while(l < r && k(q[r], q[r - 1]) >= k(q[r], i)) --r;
        q[++r] = i;
    }
    printf("%lld", f[n]);
}
```

CF455D

CF383E

CF1045G

CF797D

CF979D

CF1093G

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:farmer_john:jjleo:2020.05.16-2020.05.22&rev=1590156855

Last update: **2020/05/22 22:14**