

Crack-A-Hack

Arrays and Queries

Course: Algorithmic Problem Solving

Course Code: 17ECSE309

By:-

Aakash Huilgol

USN: 01FE15BCS002

1. Introduction

A subsequence S with length L is called *beautiful* if and only if:

- $L = 1$ or
- Let S' be a sorted version of S . It must hold that for every $S'_i = S'_{i+1} - 1$ for every $i \in [1, L-1]$.

For example:

Consider the list $[1, 2, 3, 4, 3, 5]$.

The beautiful sequences possible are: $[1, 2, 3]$ and $[4, 3, 5]$

$[1, 2, 3, 4, 5]$ and $[3]$

2. Code

```
#include <iostream>
#include <iomanip>
#include <cstdlib>
#include <algorithm>
#include <fstream>
#include <cstdio>
#include <cmath>
#include <cstring>
#include <string>
#include <ctime>
#include <queue>
#include <stack>
```

```
#include <vector>
#include <map>
#include <set>
#include <deque>
#include <cassert>
#include <unordered_map>
#include <bitset>
#include <unordered_set>
```

```
using namespace std;
```

```
#define pb push_back
#define pp pop_back
#define f first
#define s second
#define mp make_pair
#define sz(a) (int)((a).size())
#ifdef _WIN32
# define I64 "%I64d"
#else
# define I64 "%lld"
#endif
#define fname "."
```

```
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;
typedef pair < int, int > pi;
typedef pair < int, ull > pu;
typedef pair < ll, ll > pl;
```

```
const int inf = (int)1e9 + 123;
const ll infl = (ll)1e18 + 123;
const double eps = 1e-9;
```

```
const int MAX_N = (int)1e6 + 5;
const int mod = (int)1e9 + 7;
```

```
int n, q;
int a[MAX_N];
```

```
int ans;
```

```
int cnt[2 * MAX_N];
vector < int > t;
```

```
int calc(int x) {
    int id = (int)(lower_bound(t.begin(), t.end(), x) -
t.begin());
    if (id < 0 || id >= sz(t) || t[id] != x)
        return -1;
    return id;
}
```

```
void apply(int x, int val) {
    int f = calc(x - 1);
    int s = calc(x);
    int t = calc(x + 1);
    assert(s != -1);
    ans -= max(0, (f == -1 ? 0 : cnt[f]) - cnt[s]);
    ans -= max(0, cnt[s] - (t == -1 ? 0 : cnt[t]));
    cnt[s] += val;
```

```
    ans += max(0, (f == -1 ? 0 : cnt[f]) - cnt[s]);  
    ans += max(0, cnt[s] - (t == -1 ? 0 : cnt[t]));  
}
```

```
int id[MAX_N], val[MAX_N];
```

```
int main() {  
#ifdef DEBUG  
    freopen("input.txt", "r", stdin);  
#endif  
    scanf("%d", &n);  
  
    for (int i = 1; i <= n; i++) {  
        scanf("%d", &a[i]);  
        t.pb(a[i]);  
    }  
    scanf("%d", &q);  
    for (int i = 1; i <= q; i++) {  
        scanf("%d%d", &id[i], &val[i]);  
        t.pb(val[i]);  
    }  
    sort(t.begin(), t.end());  
    t.resize(unique(t.begin(), t.end()) - t.begin());  
  
    for (int i = 1; i <= n; i++) {  
        apply(a[i], 1);  
    }  
    int res = 0;  
    for (int i = 1; i <= q; i++) {  
        apply(a[id[i]], -1);  
        a[id[i]] = val[i];  
    }  
}
```

```
        apply(a[id[i]], 1);
        res = (1ll * i * ans + res) % mod;
    }
    printf("%d\n", res);
    return 0;
}
```

3. References

<https://www.hackerrank.com/contests/hackerrank-hiring-contest/challenges/array-and-queries-1/forum>

<https://stackoverflow.com/questions/3578083/what-is-the-best-way-to-use-a-hashmap-in-c>