

E2J 任务达人莫卡

题目描述

起始位置为 k ，每天有一个任务，前往 c_i 可获得 b_{c_i} ，如当前即在 c_i ，不动可获得 a_{c_i}

数据范围

$$1 \leq a_i, b_i \leq 10^9$$

n, m 都是 10^5 级别

解题思路

- 设当前状态 f_i ，表示最后一次从某个城市转移到 c_i 可以获得的最大价值
- 对于当前若在 c_i ，则一定是从最近的依然处于 c_i 的状态转移而来，这之间都选择不进行城市的变化

$$f_i = \max\{f_i, f[\text{last}[c_i]] + a_{c_i}\}$$

- 对于当前不在 c_i ，由于每次转移城市的价值都是 > 0 的，故由于本次我们一定会转移到 c_i ，则前边可转移尽量转移，由上一个状态转移过来一定最优

$$f_i = \max\{f_i, f_{i-1} + b_{c_i}\}$$

时间复杂度: $O(m)$

代码实现

```
#include <bits/stdc++.h>

using namespace std;

using ll = long long;
using db = long double; // or double, if TL is tight
using str = string;
using u32 = unsigned int;
using u64 = unsigned long long;
using u128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<db, db>;
#define fi first
#define se second
```

```

#define vc vector
using vi = vc<int>;
using vb = vc<bool>;
using vll = vc<ll>;
using vd = vc<db>;
using vs = vc<str>;
using vch = vc<char>;
using vpil = vc<pi>;
using vpll = vc<pll>;
using vpdd = vc<pdd>;
using vvi = vc<vc<int>>;
using vvll = vc<vc<ll>>;
using vvch = vc<vc<char>>;
using vvbi = vc<vc<bool>>;
using vvpi = vc<vc<pi>>;
using vvpll = vc<vc<pll>>;

const int mod = 998244353; // 1e9 + 7;
const int INF = 0x3f3f3f3f;
const ll BIG = 1e18; // not too close to LLONG_MAX
const db PI = acos((db)-1);
const int dx[4]{1, 0, -1, 0}, dy[4]{0, 1, 0, -1}; // for
every grid problem!!
mt19937
rng32((uint32_t)chrono::steady_clock::now().time_since_epo
ch().count());
mt19937_64
rng64((uint64_t)chrono::steady_clock::now().time_since_epo
ch().count());

#define cl(x, y) memset(x, y, sizeof(x))
#define de(x) cerr << #x << " = " << x << " "
#define del(x) cerr << #x << " = " << x << endl

#define overload3(a, b, c, d, ...) d
#define rep1(a) for(int i = 0; i < (a); i++)
#define rep2(i, a) for (int i = 0; i < (a); i++)
#define rep3(i, a, b) for (int i = (a); i <= (b); i++)
#define per1(a) for(int i = (a) - 1; i >= 0; i--)
#define per2(i, a) for (int i = (a) - 1; i >= 0; i--)
#define per3(i, a, b) for (int i = (a); i >= (b); i--)
#define rep(args...) overload3(args, rep3, rep2, rep1)
(args)
#define per(args...) overload3(args, per3, per2, per1)
(args)
#define for_subset(t, s) \
    for (ll t = (s); t >= 0; t = (t == 0 ? -1 : (t - 1) &
(s)))

#define co cout
#define dl '\n'

```

```

#define ret return
#define pb push_back
#define tcT template <class T
#define YES cout << "YES" << endl
#define Yes cout << "Yes" << endl
#define NO cout << "NO" << endl
#define No cout << "No" << endl
#define INT(args...) \
    int args; \
    rd(args)
#define LL(args...) \
    ll args; \
    rd(args)

#define all(s) s.begin(), s.end()
#define rall(s) s.rbegin(), s.rend()
#define sz(v) int(v.size())
#define SUM(a) accumulate(all(a), 0ll)
#define MIN(v) (*min_element(all(v)))
#define MAX(v) (*max_element(all(v)))
#define unisort(v) \
    sort(all(v)); \
    v.resize(unique(all(v)) - v.begin())

int popcnt(int x) { return __builtin_popcount(x); }
int popcnt(u32 x) { return __builtin_popcount(x); }
int popcnt(ll x) { return __builtin_popcountll(x); }
int popcnt(u64 x) { return __builtin_popcountll(x); }

#define LOWBIT(x) ((x) & (-x))
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 1, 2)
int topbit(int x) { return (x == 0 ? -1 : 31 - __builtin_clz(x)); }
int topbit(u32 x) { return (x == 0 ? -1 : 31 - __builtin_clz(x)); }
int topbit(ll x) { return (x == 0 ? -1 : 63 - __builtin_clzll(x)); }
int topbit(u64 x) { return (x == 0 ? -1 : 63 - __builtin_clzll(x)); }
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 0, 2)
int lowbit(int x) { return (x == 0 ? -1 : __builtin_ctz(x)); }
int lowbit(u32 x) { return (x == 0 ? -1 : __builtin_ctz(x)); }
int lowbit(ll x) { return (x == 0 ? -1 : __builtin_ctzll(x)); }
int lowbit(u64 x) { return (x == 0 ? -1 : __builtin_ctzll(x)); }
tcT > T cdiv(T &a, T &b) {
    return a / b + ((a ^ b) > (T)(0) && a % b);
} // divide a by b rounded up
tcT > T fdiv(T &a, T &b) {

```

```

        return a / b - ((a ^ b) < (T)(0) && a % b);
    } // divide a by b rounded down

tcT, class S > inline ostream &operator<<(ostream &i,
const pair<T, S> &a) { return i << a.first << ' ' <<
a.second; }
tcT, class S > inline istream &operator>>(istream &i,
pair<T, S> &a) { return i >> a.first >> a.second; }
tcT > inline ostream &operator<<(ostream &i, const
vector<T> &a) {
    if (a.size())
        i << a[0];
    for (int j = 1; j < int(a.size()); j++)
        i << ' ' << a[j];
    return i;
}
tcT > inline istream &operator>>(istream &i, vector<T> &a)
{
    for (auto &j : a)
        i >> j;
    return i;
}
void err(istream_iterator<string> it) {
it.~istream_iterator(); }
template <typename T, typename... Args>
void err(istream_iterator<string> it, T a, Args... args) {
    cerr << *it << " = " << a << endl;
    err(++it, args...);
}
void wt() { cout << '\n'; }
tcT, class... Args > void wt(const T &a, const Args
&...args) {
    cout << a;
    (void)(cout << ... << (cout << ' ', args));
    cout << '\n';
}
tcT > void wt(vector<vector<T>> &a) {
    for (auto &i : a)
        wt(i);
}
tcT, class S > void wt(vector<pair<T, S>> &a) {
    for (auto &i : a)
        wt(i);
}
template <class... T>
void rd(T &...a) { (cin >> ... >> a); }

#define LOCAL_TEST freopen("out.txt", "w", stdout)
#define IOS
ios::sync_with_stdio(false);cin.tie(0);cout.tie(0)
#define FIX(x) cout << fixed << setprecision(x)

```

```

const int maxn = 1e5 + 5;

inline void INIT() {

}

inline void SOLVE() {

    // J pass

    INT(n, k);

    vll a(n + 1), b(n + 1);
    rep(i, 1, n) cin >> a[i];
    rep(i, 1, n) cin >> b[i];

    INT(m);
    vll f(m + 2, -1e18), loc(n + 1, m + 1);
    f[0] = 0; loc[k] = 0;

    rep(i, 1, m) {
        INT(c);
        if(loc[c] != i - 1) f[i] = f[i - 1] + b[c];
        f[i] = max(f[i], f[loc[c]] + a[c]);
        loc[c] = i;
    }

    ll ans = 0;
    rep(i, 1, m) ans = max(ans, f[i]);
    wt(ans);

}

int main() {

    IOS; FIX(20);

    INIT();
    int tt = 1;
    cin >> tt;
    while(tt--)
        SOLVE();

    // cout << "program ends..." << endl;
    // system("pause");
    return 0;

}

```