Standard Code Library

tingyx

njust

October 2, 2023

Contents

一切的开始

Codeforces/XCPC

- 需要 C++17/C++20

```cpp
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define vi vector<int>
#define vl vector<ll>
#define x first
#define y second
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j ,k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 2e5 + 10;
const ll mod = 998244353;
const ll inf32 = 1e9;
const ll inf64 = 1e18;


void solve(){

}

int main(){
    ios;
    //freopen("sample.txt", "r", stdin);
    //freopen("resout.txt", "w", stdout);
    int t = 1;
    //cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
// --------------------------------------------------------------------------------
```

int128

- 不要使用 cin/cout，记得关同步流

```cpp
typedef __int128 i128;

i128 read()
{
   i128 x = 0; bool f = 0;
   char c = getchar();
   while (c < '0' || c > '9')
   {
       if (c == '-')
           f = 1;
       c = getchar();
   }
   while (c >= '0' && c <= '9')
   {
       x = (x << 1) + (x << 3) + (c ^ 48);
       c = getchar();
   }
   return f ? -x : x;
}

inline void write(i128 x)
{
   if (x < 0)
       putchar('-'), x = -x;
   if (x > 9)
```

```
        write(x / 10);
    putchar(x % 10 + '0');
}
```

数据结构

二维数点

- 逆序对

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn = 500010;
ll m;
ll a[maxn], b[maxn], c[maxn];
int lowbit(int x){return x & (-x);}
void add(int x, ll y){
    for (int i = x; i <= m; i += lowbit(i)) c[i] += y;
}
ll sum(int x){
    ll res = 0;
    for (int i = x; i; i -= lowbit(i)) res += c[i];
    return res;
}
int main(){
    int n;
    cin >> n;
    for (int i = 1; i <= n; ++i){
        cin >> a[i];
        b[i] = a[i];
    }
    sort(b + 1, b + n + 1);
    m = unique(b + 1, b + n + 1) - b - 1;
    ll ans = 0;
    for (int i = n; i; i--){
        int k = lower_bound(b + 1, b + m + 1, a[i]) - b;
        ans += sum(k - 1);
        add(k, 1);
    }
    cout << ans;
    return 0;
}
```

- 园丁的烦恼 (矩阵内点的个数)

```
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define pii pair<int, int>
#define vi vector<int>
#define vl vector<ll>
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j ,k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 1e7 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

struct BIT{
    int tr[maxn];
    int lowbit(int x){return x & -x;}
    void add(int p, int x){
        for (; p < maxn; p += lowbit(p)) tr[p] += x;
    }
    ll query(int p){
        ll sum = 0;
```

```
        for (; p > 0; p -= lowbit(p))
            sum += tr[p];
        return sum;
    }
}Tr;

void solve(){
    int n, m;
    cin >> n >> m;
    vector<pii> pos;
    vector<tuple<int, int, int, int>> q;
    vector<ll> ans(m + 1);
    rep(i, 1, n){
        int tx, ty;
        cin >> tx >> ty;
        tx++, ty++;
        pos.push_back({tx, ty});
    }
    sort(pos.begin(), pos.end());
    rep(i, 1, m){
        int x1, y1, x2, y2;
        cin >> x1 >> y1 >> x2 >> y2;
        x1++, y1++, x2++, y2++;
        q.push_back({x1 - 1, y1 - 1, 1, i});
        q.push_back({x1 - 1, y2, -1, i});
        q.push_back({x2, y1 - 1, -1, i});
        q.push_back({x2, y2, 1, i});
    }
    sort(q.begin(), q.end());
    int cur = 0;
    for (auto [x, y, c, id] : q){
        while (cur < n && pos[cur].first <= x) Tr.add(pos[cur++].second, 1);
        ans[id] += c * Tr.query(y);
    }
    rep(i, 1, m) cout << ans[i] << endl;
}

int main(){
    ios;
    //freopen("sample.txt", "r", stdin);
    //freopen("resout.txt", "w", stdout);
    int t = 1;
    //cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
```

- HH 的项链（区间元素种类）照常把 x 所在一维降掉后，发现 y 轴并没有明显的偏序关系。可以这样考虑，我们只计每个元素第一次在区间中出现时有贡献，设 pre[i] 表示位置 i 的元素前一次出现的位置，在整个序列中第一次出现时记为 0

```
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define vi vector<int>
#define vl vector<ll>
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j ,k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 1e6 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

struct BIT{
    ll tr[maxn];
    int lowbit(int x){return x & -x;}
    void add(int p, ll x){
```

```
        for (; p < maxn; p += lowbit(p)) tr[p] += x;
    }
    ll query(int p){
        ll sum = 0;
        for (; p > 0; p -= lowbit(p))
            sum += tr[p];
        return sum;
    }
}Tr;

ll pre[maxn], ans[maxn];
void solve(){
    int n, m;
    cin >> n;
    vector<pll> pos;
    vector<tuple<int, int, int, int>> q;
    for (int i = 3; i <= n + 2; ++i){
        int a;
        cin >> a;
        pos.push_back({i, pre[a] ? pre[a] : 2}), pre[a] = i;
    }
    sort(pos.begin(), pos.end());
    cin >> m;
    for (int i = 1; i <= m; ++i){
        int l, r;
        cin >> l >> r;
        l += 2, r += 2;
        q.push_back({l - 1, 1, 1, i});
        q.push_back({l - 1, l - 1, -1, i});
        q.push_back({r, 1, -1, i});
        q.push_back({r, l - 1, 1, i});
    }
    sort(q.begin(), q.end());
    int cur = 0;
    for (auto [x, y, c, id] : q)
    {
        while (cur < n && pos[cur].first <= x)
            Tr.add(pos[cur++].second, 1);
        ans[id] += c * Tr.query(y);
    }
    for (int i = 1; i <= m; i++) cout << ans[i] << endl;
}

int main(){
    ios;
    //freopen("sample.txt", "r", stdin);
    //freopen("resout.txt", "w", stdout);
    int t = 1;
    //cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
```

- 矩阵内权值之和

```
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define vi vector<int>
#define vl vector<ll>
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j ,k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 3e5 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

```

```cpp
struct BIT{
    ll tr[maxn];
    int lowbit(int x){return x & -x;}
    void add(int p, ll x){
        for (; p < maxn; p += lowbit(p)) tr[p] += x;
    }
    ll query(int p){
        ll sum = 0;
        for (; p > 0; p -= lowbit(p))
            sum += tr[p];
        return sum;
    }
}Tr;

void solve(){
    int n, m;
    cin >> n >> m;
    vector<tuple<int, int, int>> pos;
    vector<tuple<int, int, int, int>> q;
    vector<ll> ans(m + 1);
    vector<int> yy;
    rep(i, 1, n){
        int x, y, p;
        cin >> x >> y >> p;
        yy.push_back(y);
        pos.push_back({x, y, p});
    }
    sort(pos.begin(), pos.end());
    rep(i, 1, m){
        int x1, y1, x2, y2;
        cin >> x1 >> y1 >> x2 >> y2;
        yy.push_back(y1 - 1), yy.push_back(y2);
        q.push_back({x1 - 1, y1 - 1, 1, i});
        q.push_back({x2, y1 - 1, -1, i});
        q.push_back({x1 - 1, y2, -1, i});
        q.push_back({x2, y2, 1, i});
    }
    sort(q.begin(), q.end());
    sort(yy.begin(), yy.end());
    yy.erase(unique(yy.begin(), yy.end()), yy.end());
    int cur = 0;
    for (auto [x, y, c, id] : q){
        y = lower_bound(yy.begin(), yy.end(), y) - yy.begin() + 1;
        while (cur < n){
            auto [_x, _y, p] = pos[cur];
            if (_x > x) break;
            _y = lower_bound(yy.begin(), yy.end(), _y) - yy.begin() + 1;
            Tr.add(_y, p), ++cur;
        }
        ans[id] += c * Tr.query(y);
    }
    for (int i = 1; i <= m; ++i) cout << ans[i] << endl;
}

int main(){
    ios;
    //freopen("sample.txt", "r", stdin);
    //freopen("resout.txt", "w", stdout);
    int t = 1;
    //cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
```

区间问题

莫队

- 区间取两个数相同概率

```cpp
#include <algorithm>
#include <cmath>
#include <cstdio>
using namespace std;
const int N = 50005;
int n, m, maxn;
int c[N];
long long sum;
int cnt[N];
long long ans1[N], ans2[N];

struct query {
  int l, r, id;

  bool operator<(const query &x) const {  // 重载 < 运算符
    if (l / maxn != x.l / maxn) return l < x.l;
    return (l / maxn) & 1 ? r < x.r : r > x.r;
  }
} a[N];

void add(int i) {
  sum += cnt[i];
  cnt[i]++;
}

void del(int i) {
  cnt[i]--;
  sum -= cnt[i];
}

long long gcd(long long a, long long b) { return b ? gcd(b, a % b) : a; }

int main() {
  scanf("%d%d", &n, &m);
  maxn = sqrt(n);
  for (int i = 1; i <= n; i++) scanf("%d", &c[i]);
  for (int i = 0; i < m; i++) scanf("%d%d", &a[i].l, &a[i].r), a[i].id = i;
  sort(a, a + m);
  for (int i = 0, l = 1, r = 0; i < m; i++) {  // 具体实现
    if (a[i].l == a[i].r) {
      ans1[a[i].id] = 0, ans2[a[i].id] = 1;
      continue;
    }
    while (l > a[i].l) add(c[--l]);
    while (r < a[i].r) add(c[++r]);
    while (l < a[i].l) del(c[l++]);
    while (r > a[i].r) del(c[r--]);
    ans1[a[i].id] = sum;
    ans2[a[i].id] = (long long)(r - l + 1) * (r - l) / 2;
  }
  for (int i = 0; i < m; i++) {
    if (ans1[i] != 0) {
      long long g = gcd(ans1[i], ans2[i]);
      ans1[i] /= g, ans2[i] /= g;
    } else
      ans2[i] = 1;
    printf("%lld/%lld\n", ans1[i], ans2[i]);
  }
  return 0;
}
```

CDQ

- 逆序对

```cpp
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define vi vector<int>
#define vl vector<ll>
#define x first
#define y second
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j ,k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 2e5 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

void solve(){
    int n;
    cin >> n;
    vi a(n + 1), temp(n + 1);
    ll ans = 0;
    rep(i, 1, n) cin >> a[i];
    function<void(int, int)> cdq = [&](int l, int r){
        if (l == r) return;
        int mid = l + r >> 1;
        cdq(l, mid);
        cdq(mid + 1, r);
        int p1 = l, p2 = mid + 1, idx = l;
        while (p1 <= mid && p2 <= r){
            if (a[p1] > a[p2]) temp[idx++] = a[p1++];
            else temp[idx++] = a[p2++], ans += p1 - l;
        }
        while (p1 <= mid) temp[idx++] = a[p1++];
        while (p2 <= r) temp[idx++] = a[p2++], ans += p1 - l;
        for (int i = l; i <= r; ++i) a[i] = temp[i];
    };
    cdq(1, n);
    cout << ans << endl;
}

int main(){
    ios;
    //freopen("sample.txt", "r", stdin);
    //freopen("resout.txt", "w", stdout);
    int t = 1;
    //cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
```

- 求最长不上升子序列和最长上升子序列

```cpp
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 100005;
int n, x, dp[MAXN], a[MAXN], ans;
pair<int, int> temp[MAXN][20]; //val, pos

bool cmp(const pair<int, int> &A, const pair<int, int> &B, const int &type) {
    return type ? A.first != B.first ? A.first > B.first : A.second < B.second : A.first != B.first ? A.first < B.first: A.second > B.second;
}

void mergeSort(int l, int r, int deep, const int &cmptype) {
    if (l == r) {
        temp[l][deep].first = a[l];
        temp[l][deep].second = l;
        return;
    }
```

```
    int mid = (l + r) >> 1;
    mergeSort(l, mid, deep + 1, cmptype);
    mergeSort(mid + 1, r, deep + 1, cmptype);
    int p1 = l, p2 = mid + 1;
    while (p1 <= mid && p2 <= r) {
        if (cmp(temp[p1][deep + 1], temp[p2][deep + 1], cmptype)) {
            temp[l++][deep] = temp[p1++][deep + 1];
        } else {
            temp[l++][deep] = temp[p2++][deep + 1];
        }
    }
    while (p1 <= mid) {
        temp[l++][deep] = temp[p1++][deep + 1];
    }
    while (p2 <= r) {
        temp[l++][deep] = temp[p2++][deep + 1];
    }
}

void cdqDivAlgorithm(int l, int r, int deep, const int &cmptype) {
    if (l == r) {
        dp[l] = max(dp[l], 1);
        ans = max(ans, dp[l]);
        return;
    }
    int mid = (l + r) >> 1;
    cdqDivAlgorithm(l, mid, deep + 1, cmptype);
    int p1 = l, p2 = mid + 1, premax = 0;
    while (p1 <= mid && p2 <= r) {
        if (cmp(temp[p1][deep + 1], temp[p2][deep + 1], cmptype)) {
            premax = max(premax, dp[temp[p1++][deep + 1].second]);
        } else {
            dp[temp[p2][deep + 1].second] = max(premax + 1, dp[temp[p2][deep + 1].second]);
            p2++;
        }
    }
    while (p2 <= r) {
        dp[temp[p2][deep + 1].second] = max(premax + 1, dp[temp[p2][deep + 1].second]);
        p2++;
    }
    cdqDivAlgorithm(mid + 1, r, deep + 1, cmptype);
}

int main()
{
    while (scanf("%d", &x) != EOF)a[++n] = x;
    mergeSort(1, n, 0, 1);
    cdqDivAlgorithm(1, n, 0, 1);
    printf("%d\n", ans);
    memset(dp, 0, sizeof(dp));
    ans = 0;
    mergeSort(1, n, 0, 0);
    cdqDivAlgorithm(1, n, 0, 0);
    printf("%d\n", ans);
    return 0;
}
```

- 求地毯覆盖（最多取多少个不相互覆盖）

```
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 1000005;
int n, L[MAXN], R[MAXN], id[MAXN], dp[MAXN], ans;
int temp[MAXN];
void cdqDivAlgorithm(int l, int r) {
    if (l == r) {
        dp[id[l]] = max(1, dp[id[l]]);
        ans = max(ans, dp[id[l]]);
        return;
    }
    int mid = (l + r) >> 1;
    cdqDivAlgorithm(l, mid);
```

```
    int p1 = l, pl, p2 = mid + 1, premax = 0;
    while (p1 <= mid && p2 <= r) {
        if (R[id[p1]] <= L[id[p2]]) {
            premax = max(premax, dp[id[p1++]]);
        } else {
            dp[id[p2]] = max(premax + 1, dp[id[p2]]);
            ++p2;
        }
    }
    while (p2 <= r) {
        dp[id[p2]] = max(premax + 1, dp[id[p2]]);
        ++p2;
    }
    cdqDivAlgorithm(mid + 1, r);
    p1 = l, pl = l, p2 = mid + 1;
    while (p1 <= mid && p2 <= r) {
        if (R[id[p1]] < R[id[p2]]) {
            temp[pl++] = id[p1++];
        } else {
            temp[pl++] = id[p2++];
        }
    }
    while (p1 <= mid) {
        temp[pl++] = id[p1++];
    }
    while (p2 <= r) {
        temp[pl++] = id[p2++];
    }
    for (int i = l; i <= r; ++i) {
        id[i] = temp[i];
    }
}
int main()
{
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        scanf("%d %d", &L[i], &R[i]);
        id[i] = i;
    }
    sort(id + 1, id + 1 + n, [](const int &A, const int &B) {
        return L[A] < L[B];
    });
    cdqDivAlgorithm(1, n);
    printf("%d\n", ans);
    return 0;
}
```

- 动态凸包

第一行：一个整数 N，表示方案和询问的总数。接下来 N 行，每行开头一个单词 “Query” 或 “Project”。若单词为 Query，则后接一个整数 T，表示 Blue Mary 询问第 T 天的最大收益。若单词为 Project，则后接两个实数 S，P，表示该种设计方案第一天的收益 S，以及以后每天比上一天多出的收益 P。对于每一个 Query，输出一个整数，表示询问的答案，并精确到整百元 1 <= N <= 100000 1 <= T <=50000 0 < P < 100，|S| <= 10^6

```
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 100005;
const double eps = 1e-6;
int m, n, id[MAXN], qid[MAXN], type[MAXN], x[MAXN], temp[MAXN], top;
double k[MAXN], b[MAXN], ans[MAXN];
char op[55];
inline bool cmp(const int &A, const int &B) {
    return type[A] != type[B] ? type[A] < type[B] : type[A] ? x[A] < x[B] : k[A] < k[B];
}
inline int dcmp(double x) {
    return x > eps ? 1 : x < -eps ? -1 : 0;
}
inline double getCross(const double &k1, const double &b1, const double &k2, const double &b2) {
    return (b2 - b1) / (k1 - k2);
}
inline double getVal(const double &k, const double &b, const int &x)
```

```
{
    return k * x + b;
}
pair<double, double>stk[MAXN];
void stkClear() {
    top = 0;
    stk[++top] = make_pair(0, 0);
}
void stkInsert(double k, double b) {
    if (dcmp(stk[top].first - k) == 0 && dcmp(stk[top].second - b) < 0)top--;
    if (dcmp(stk[top].first - k) == 0 && dcmp(stk[top].second - b) >= 0)return;
    while (top >= 2 && dcmp(getCross(stk[top].first, stk[top].second, stk[top - 1].first, stk[top - 1].second) - getCross(stk[top].first, stk[top].second, k, b)) > 0)top--;
    stk[++top] = make_pair(k, b);
}
double stkQuery(int x) {
    while (top >= 2 && dcmp(getVal(stk[top].first, stk[top].second, x) - getVal(stk[top - 1].first, stk[top - 1].second, x)) < 0)--top;
    return getVal(stk[top].first, stk[top].second, x);
}
void cdqDivAlgorithm(int l, int r) {
    if (l == r)return;
    int mid = (l + r) >> 1;
    cdqDivAlgorithm(l, mid);
    cdqDivAlgorithm(mid + 1, r);
    stkClear();
    for (int i = l; i <= mid && !type[id[i]]; ++i) {
        stkInsert(k[id[i]], b[id[i]]);
    }
    for (int i = r; i > mid && type[id[i]]; --i) {
        ans[qid[id[i]]] = max(ans[qid[id[i]]], stkQuery(x[id[i]]));
    }
    int p1 = l, pl = l, p2 = mid + 1;
    while (p1 <= mid && p2 <= r) {
        if (cmp(id[p1], id[p2])) {
            temp[pl++] = id[p1++];
        } else {
            temp[pl++] = id[p2++];
        }
    }
    while (p1 <= mid) {
        temp[pl++] = id[p1++];
    }
    while (p2 <= r) {
        temp[pl++] = id[p2++];
    }
    for (int i = l; i <= r; ++i) {
        id[i] = temp[i];
    }
}
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        id[i] = i;
        scanf("%s", op);
        if (*op == 'P') {
            type[i] = 0;
            scanf("%lf %lf", &b[i], &k[i]);
            b[i] -= k[i];
        }
        else {
            type[i] = 1;
            qid[i] = ++m;
            scanf("%d", &x[i]);
        }
    }
    cdqDivAlgorithm(1, n);
    for (int i = 1; i <= m; ++i) {
        printf("%d\n", (int)ans[i] / 100);
    }
    return 0;
```

```
}

```

树上问题

树剖

- 2018ICPC 青岛网络赛（多测时候用来剖的）

```cpp
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define vi vector<int>
#define vl vector<ll>
#define x first
#define y second
#define rep(i, j, k) for(int i = (j); i <= (k); i++)
#define per(i, j, k) for(int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 1e5 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

void solve()
{
    int n, m, q, k, cnt = 0;
    cin >> n >> m >> q;
    vi red(n + 1);
    vector<vector<pll>> G(n + 1);
    vl dis(n + 1), dep(n + 1), v(n + 1);
    vi dfn(n + 1), idx(n + 1);
    vi son(n + 1, -1), sz(n + 1), fa(n + 1), top(n + 1);
    function<void(int, int)> dfs1 = [&](int u, int f) {
        son[u] = -1;
        sz[u] = 1;
        if(!red[u])
            red[u] = red[f];
        for(auto [v, w] : G[u]) {
            if(v == f)
                continue;
            dep[v] = dep[u] + 1;
            dis[v] = dis[u] + w;
            fa[v] = u;
            dfs1(v, u);
            sz[u] += sz[v];
            if(son[u] == -1 || sz[v] > sz[son[u]])
                son[u] = v;
        }
    };
    function<void(int, int)> dfs2 = [&](int u, int t) {
        top[u] = t;
        dfn[u] = ++cnt;
        idx[cnt] = u;
        if(son[u] == -1)
            return;
        dfs2(son[u], t);
        for(auto [v, w] : G[u])
            if(v != son[u] && v != fa[u])
                dfs2(v, v);
    };
    auto lca = [&](int u, int v) {
        while(top[u] != top[v]) {
            if(dep[top[u]] > dep[top[v]])
                u = fa[top[u]];
            else
                v = fa[top[v]];
        }
```

```
        return dep[u] > dep[v] ? v : u;
    };
    for(int i = 1, x; i <= m; ++i)
        cin >> x, red[x] = x;
    for(int i = 1; i < n; ++i) {
        int u, v, w;
        cin >> u >> v >> w;
        G[u].push_back({v, w});
        G[v].push_back({u, w});
    }
    dfs1(1, 0);
    dfs2(1, 1);
    for(int i = 1; i <= n; ++i)
        v[i] = dis[i] - dis[red[i]];
    while(q--) {
        cin >> k;
        vector<int> p(k + 1);
        auto check = [&](ll st) {
            vector<int> q;
            for(int i = 1; i <= k; ++i)
                if(v[p[i]] > st)
                    q.push_back(p[i]);
            if(q.size() == 0)
                return true;
            int mnd = n + 1, mxd = 0;
            for(int i = 0; i < q.size(); ++i) {
                mnd = min(mnd, dfn[q[i]]);
                mxd = max(mxd, dfn[q[i]]);
            }
            int ca = lca(idx[mnd], idx[mxd]);
            for(int i = 0; i < q.size(); ++i)
                if(dis[q[i]] - dis[ca] > st)
                    return false;
            return true;
        };
        ll mx = 0;
        for(int i = 1; i <= k; ++i) {
            cin >> p[i];
            mx = max(mx, v[p[i]]);
        }
        ll l = 0, r = mx;
        while(l < r) {
            ll mid = (l + r) >> 1;
            if(check(mid))
                r = mid;
            else
                l = mid + 1;
        }
        cout << l << endl;
    }
}

int main()
{
    ios;
    // freopen("sample.txt", "r", stdin);
    // freopen("resout.txt", "w", stdout);
    int t = 1;
    cin >> t;
    while(t--) {
        solve();
    }
    return 0;
}

```

计算几何

二维几何：点与向量

```cpp
#define y1 yy1
#define nxt(i) ((i + 1) % s.size())
typedef double LD;
const LD PI = 3.14159265358979323846;
const LD eps = 1E-10;
int sgn(LD x) { return fabs(x) < eps ? 0 : (x > 0 ? 1 : -1); }
struct L;
struct P;
typedef P V;
struct P {
    LD x, y;
    explicit P(LD x = 0, LD y = 0): x(x), y(y) {}
    explicit P(const L& l);
};
struct L {
    P s, t;
    L() {}
    L(P s, P t): s(s), t(t) {}
};

P operator + (const P& a, const P& b) { return P(a.x + b.x, a.y + b.y); }
P operator - (const P& a, const P& b) { return P(a.x - b.x, a.y - b.y); }
P operator * (const P& a, LD k) { return P(a.x * k, a.y * k); }
P operator / (const P& a, LD k) { return P(a.x / k, a.y / k); }
inline bool operator < (const P& a, const P& b) {
    return sgn(a.x - b.x) < 0 || (sgn(a.x - b.x) == 0 && sgn(a.y - b.y) < 0);
}
bool operator == (const P& a, const P& b) { return !sgn(a.x - b.x) && !sgn(a.y - b.y); }
P::P(const L& l) { *this = l.t - l.s; }
ostream &operator << (ostream &os, const P &p) {
    return (os << "(" << p.x << "," << p.y << ")");
}
istream &operator >> (istream &is, P &p) {
    return (is >> p.x >> p.y);
}

LD dist(const P& p) { return sqrt(p.x * p.x + p.y * p.y); }
LD dot(const V& a, const V& b) { return a.x * b.x + a.y * b.y; }
LD det(const V& a, const V& b) { return a.x * b.y - a.y * b.x; }
LD cross(const P& s, const P& t, const P& o = P()) { return det(s - o, t - o); }
// ------------------------------------------------
```

字符串

KMP

- KMP 模板

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

using namespace std;

const int N = 1e6 + 10;

vector<int> prefix_function(string s)
{
    int n = (int)s.length();
    vector<int> pi(n);
    for (int i = 2; i < n; i++)
    {
        pi[i] = pi[i - 1];
        while (pi[i] && s[i] != s[pi[i] + 1])
            pi[i] = pi[pi[i]];
        pi[i] += (s[i] == s[pi[i] + 1]);
    }
```

```
    return pi;
}

int main(void)
{
    ios::sync_with_stdio(false), cin.tie(0), cout.tie(0);
    string s1, s2;
    cin >> s1 >> s2;
    s1 = " " + s1;
    s2 = " " + s2;
    auto nxt = prefix_function(s2);
    for (int i = 1, j = 0; i < s1.size(); i++)
    {
        while (j && s1[i] != s2[j + 1])
            j = nxt[j];
        if (s1[i] == s2[j + 1])
            j++;
        if (j == s2.size() - 1)
        {
            cout << i - j + 1 << "\n";
            j = nxt[j];
        }
    }
    for (int i = 1; i < s2.size(); i++)
        cout << nxt[i] << " ";

    return 0;
}
```

- carpet(二维 KMP) 有一个 n*m 的地毯，aij 表示地毯每格的元素，bij 表示地毯每格的价格，要求选取一块价格最大值最小的地毯，并且这块地毯无限铺开之后，原地毯是其子矩阵

```
#include <bits/stdc++.h>
#define endl '\n'
#define pll pair<ll, ll>
#define tll tuple<ll, ll, ll>
#define x first
#define y second
#define int ll
#define rep(i, j, k) for (int i = (j); i <= (k); i++)
#define per(i, j, k) for (int i = (j); i >= (k); i--)
#define ios ios::sync_with_stdio(false), cin.tie(0), cout.tie(0)
using namespace std;
typedef long long ll;
const ll maxn = 1e6 + 10;
const ll mod = 998244353;
const ll inf = 0x3f3f3f3f;

vector<int> prefix_function(string s)
{
    int n = (int)s.length();
    vector<int> pi(n);
    for (int i = 2; i < n; i++)
    {
        pi[i] = pi[i - 1];
        while (pi[i] && s[i] != s[pi[i] + 1])
            pi[i] = pi[pi[i]];
        pi[i] += (s[i] == s[pi[i] + 1]);
    }
    return pi;
}

int get_length(vector<string> s)
{
    int len = s[1].size() - 1;
    int ret = len;
    vector<int> cnt(len + 1);
    for (int i = 1; i < s.size(); ++i)
    {
        string tmp = s[i];
        auto nxt = prefix_function(tmp);
```

```
        int j = len;
        while (j)
        {
            cnt[len - nxt[j]]++;
            j = nxt[j];
        }
    }
    for (int i = 1; i <= len; ++i)
        if (cnt[i] == s.size() - 1)
        {
            ret = i;
            break;
        }
    return ret;
}

void solve()
{
    int n, m;
    cin >> n >> m;
    vector<string> s1(n + 1);
    for (int i = 1; i <= n; ++i)
        cin >> s1[i], s1[i] = " " + s1[i];
    vector<string> s2(m + 1);
    for (int i = 1; i <= m; ++i)
    {
        string tmp = " ";
        for (int j = 1; j <= n; ++j)
            tmp += s1[j][i];
        s2[i] = tmp;
    }
    vector<vector<int>> a(n + 1, vector<int>(m + 1, 0));
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= m; ++j)
            cin >> a[i][j];
    int p = get_length(s1), q = get_length(s2);
    ll ans = 1e9;
    deque<int> dq;
    auto b = a;
    for (int i = 1; i <= n; ++i){
        while (dq.size()) dq.pop_back();
        for (int j = 1; j <= m; ++j){
            while (dq.size() && j - dq.front() + 1 > p) dq.pop_front();
            while (dq.size() && a[i][dq.back()] <= a[i][j]) dq.pop_back();
            dq.push_back(j);
            b[i][j] = a[i][dq.front()];
        }
    }
    for (int j = 1; j <= m; ++j){
        while (dq.size()) dq.pop_back();
        for (int i = 1; i <= n; ++i){
            while (dq.size() && i - dq.front() + 1 > q) dq.pop_front();
            while (dq.size() && b[dq.back()][j] <= b[i][j]) dq.pop_back();
            dq.push_back(i);
            if (i >= q && j >= p)
                ans = min(ans, 1ll * b[dq.front()][j]);
        }
    }
    ans = ans * (p + 1) * (q + 1);
    cout << ans << endl;
}

signed main()
{
    ios;
    // freopen("sample.txt", "r", stdin);
    // freopen("resout.txt", "w", stdout);
    int t = 1;
    // cin >> t;
    while (t--)
    {
```

```
        solve();
    }
    return 0;
}

```

杂项

STL

- copy

```
template <class InputIterator, class OutputIterator>
  OutputIterator copy (InputIterator first, InputIterator last, OutputIterator result);
```