

`__int128`

01trie

埃氏筛

欧拉筛

DSU

大数

笛卡尔树

二维前缀和

高斯约旦消元法(double)

换根DP

计算几何_dyh

快速幂

莫队分块

三点求圆心坐标

树的直径

fenwick_tree

数位DP递推

数位DP记忆化搜索

topsort

exgcd

inverse

组合数预处理

最小球覆盖_模拟退火

最小圆覆盖

最大最小表示法(循环同构字符串)

BIT_difference

BM递推_dyh(找规律)

Chothlly_Tree区间和

CRT

Deque_手写
dijkstra_ $O(n^2)$
dijkstra_vis
dijkstra
dijkstra_path
Dinic
DinicOjWiki
Hash_string
Hash手写
kmp
LCA
Lucas
manacher
merge_sort
Miller_Rabbin
segment_tree 区间修改区间和
SegTree 区间max单点修改
SegTree 区间min单点修改
ST表
tarjan求割边
Tarjan求割点
zkw线段树
scaning_line_area
主席树_昆明M
主席树_区间第k小
主席树_区间前K大的和

```
1  /*
2      Don't forget to close:
3          ios::sync_with_stdio(false);
4          cin.tie(nullptr);
5  */
6  inline __int128 read() {
7      __int128 x = 0, f = 1;
```

```

8     char ch = getchar();
9     while (ch < '0' || ch > '9') {if (ch == '-') f = -1; ch = getchar(); }
10    while (ch >= '0' && ch <= '9') {x = x * 10 + ch - '0'; ch = getchar();
11    }
12    return x * f;
13    }
14
15    inline void print(__int128 x) {
16        if (x < 0) putchar('-'), x = -x;
17        if (x > 9) print(x / 10);
18        putchar(x % 10 + '0');
19    }
20
21    template<typename T> inline void read(T &x) {
22        x = 0;
23        T f = 1;
24        char ch = getchar();
25        while (!isdigit(ch)) {if (ch == '-') f = -1; ch = getchar(); }
26        while (isdigit(ch)) {x = x * 10 + ch - '0'; ch = getchar(); }
27        x *= f;
28    }
29
30    template<typename T> inline void write(T x) {
31        if (x < 0) putchar('-'), x = -x;
32        if (x > 9) write(x / 10);
33        putchar(x % 10 + '0');
34    }
35
36
37
38
39    namespace FastIO {
40        const int SIZE = 1 << 16;
41        char buf[SIZE], str[64];
42        int l = SIZE, r = SIZE;
43        int read(char *s) {
44            while (r) {
45                for (; l < r && buf[l] <= ' '; l++);
46                if (l < r) break;
47                l = 0, r = int(fread(buf, 1, SIZE, stdin));
48            }
49            int cur = 0;
50            while (r) {
51                for (; l < r && buf[l] > ' '; l++) s[cur++] = buf[l];
52                if (l < r) break;
53                l = 0, r = int(fread(buf, 1, SIZE, stdin));
54            }
55            s[cur] = '\0';

```

```

56         return cur;
57     }
58     template<typename type>
59     bool read(type &x, int len = 0, int cur = 0, bool flag = false) {
60         if (!(len = read(str))) return false;
61         if (str[cur] == '-') flag = true, cur++;
62         for (x = 0; cur < len; cur++) x = x * 10 + str[cur] - '0';
63         if (flag) x = -x;
64         return true;
65     }
66     template <typename type>
67     type read(int len = 0, int cur = 0, bool flag = false, type x = 0) {
68         if (!(len = read(str))) return false;
69         if (str[cur] == '-') flag = true, cur++;
70         for (x = 0; cur < len; cur++) x = x * 10 + str[cur] - '0';
71         return flag ? -x : x;
72     }
73 } using FastIO::read;

```

```

1  /*
2     Don't forget to init()!
3  */
4  struct Trie {
5      static constexpr int MAXBIT = 30;
6      int next[N][2], cnt, num[N], sz[N][2];
7
8      void init() {
9          memset(next, 0, sizeof(next));
10         memset(num, 0, sizeof(num));
11         memset(sz, 0, sizeof(sz));
12         cnt = 0;
13     }
14
15     void insert(int x) {
16         int cur = 0;
17         for (int i = MAXBIT; ~i; i--) {
18             int bit = (x >> i) & 1;
19             if (!next[cur][bit]) next[cur][bit] = ++cnt;
20             sz[cur][bit]++;
21             cur = next[cur][bit];
22         }
23         num[cur] = x;
24     }
25
26     int query(int x) {
27         int cur = 0, ans = 0;
28         for (int i = MAXBIT; ~i; i--) {
29             int bit = (x >> i) & 1;

```

```

30         if (sz[cur][bit] < (1 << i)) {
31             cur = next[cur][bit];
32         } else {
33             cur = next[cur][bit ^ 1];
34             ans |= 1 << i;
35         }
36         if (!cur) break;
37     }
38     return ans;
39 }
40 } trie;

```

```

1  /*
2   O(NloglogN)
3  */
4  vector<bool> isprime;
5  vector<int> prime;
6
7  void getPrime(int n) {
8      isprime.assign(n + 1, true);
9      isprime[0] = isprime[1] = false;
10     prime.clear();
11     for (int i = 2; i <= n; i++) {
12         if (!isprime[i]) continue;
13         prime.push_back(i)
14         for (int j = i * i; j <= n; j += i) isprime[j] = false;
15     }
16 }
17

```

```

1  vector<int> primes;
2  vector<bool> isprime;
3
4  void getPrime(int n){
5      isprime.assign(n + 1, true);
6      isprime[0] = isprime[1] = false;
7      for (int i = 2; i <= n; ++i) {
8          if (isprime[i]) primes.emplace_back(i);
9          for (auto p : primes) {
10              if (i * p > n) break;
11              isprime[i * p] = false;
12              if (i % p == 0) break;
13          }
14      }
15 }

```

```

1  class dsu {

```

```

2 public:
3     vector<int> p;
4     int n;
5
6     dsu(int _n) : n(_n) {
7         p.resize(n);
8         iota(p.begin(), p.end(), 0);
9     }
10
11     inline int get(int x) {
12         return (x == p[x] ? x : (p[x] = get(p[x])));
13     }
14
15     inline bool unite(int x, int y) {
16         x = get(x);
17         y = get(y);
18         if (x != y) {
19             p[x] = y;
20             return true;
21         }
22         return false;
23     }
24 };
25
26
27
28
29 struct DSU {
30     std::vector<int> f, siz;
31     std::vector<bool> g;
32     DSU(int n) : f(n), siz(n, 1), g(n) { std::iota(f.begin(), f.end(), 0); }
33
34     int leader(int x) {
35         while (x != f[x]) x = f[x] = f[f[x]];
36         return x;
37     }
38     bool same(int x, int y) { return leader(x) == leader(y); }
39     bool merge(int x, int y) {
40         x = leader(x);
41         y = leader(y);
42         if (x == y) return false;
43         if (g[x] && g[y]) {
44             return false;
45         }
46         siz[x] += siz[y];
47         g[x] = g[x] || g[y];
48         f[y] = x;
49         return true;
50     }
51 }

```

```

50     int size(int x) { return siz[leader(x)]; }
51 };
52
53
54
55
56 int f[N], deep[N];
57
58 void init(int x) { for (int i = 0; i <= x; i++) f[i] = i; }
59
60 int find(int x) { return f[x] == x ? x : f[x] = find(f[x]); } //路径压缩
61
62 inline void merge(int x, int y) {
63     int u = find(x), v = find(y);
64     if (u != v) {
65         if (deep[u] > deep[v]) swap(u, v);
66         f[u] = v;
67         if (deep[u] == deep[v]) deep[v]++;
68     }
69 }

```

```

1  int max(int a, int b) { return a>b?a:b; }
2  struct bign {
3      int len, s[numlen];
4      bign() {
5          memset(s, 0, sizeof(s));
6          len = 1;
7      }
8      bign(int num) { *this = num; }
9      bign(const char *num) { *this = num; }
10     bign operator = (const int num) {
11         char s[numlen];
12         sprintf(s, "%d", num);
13         *this = s;
14         return *this;
15     }
16     bign operator = (const char *num) {
17         len = strlen(num);
18         while(len > 1 && num[0] == '0') num++, len--;
19         for(int i = 0; i < len; i++) s[i] = num[len-i-1] - '0';
20         return *this;
21     }
22
23     void deal() {
24         while(len > 1 && !s[len-1]) len--;
25     }
26
27     bign operator + (const bign &a) const {

```

```

28     bign ret;
29     ret.len = 0;
30     int top = max(len, a.len) , add = 0;
31     for(int i = 0; add || i < top; i++) {
32         int now = add;
33         if(i < len) now += s[i];
34         if(i < a.len) now += a.s[i];
35         ret.s[ret.len++] = now%10;
36         add = now/10;
37     }
38     return ret;
39 }
40 bign operator - (const bign &a) const {
41     bign ret;
42     ret.len = 0;
43     int cal = 0;
44     for(int i = 0; i < len; i++) {
45         int now = s[i] - cal;
46         if(i < a.len) now -= a.s[i];
47         if(now >= 0) cal = 0;
48         else {
49             cal = 1; now += 10;
50         }
51         ret.s[ret.len++] = now;
52     }
53     ret.deal();
54     return ret;
55 }
56 bign operator * (const bign &a) const {
57     bign ret;
58     ret.len = len + a.len;
59     for(int i = 0; i < len; i++) {
60         for(int j = 0; j < a.len; j++)
61             ret.s[i+j] += s[i]*a.s[j];
62     }
63     for(int i = 0; i < ret.len; i++) {
64         ret.s[i+1] += ret.s[i]/10;
65         ret.s[i] %= 10;
66     }
67     ret.deal();
68     return ret;
69 }
70
71 //乘以小数，直接乘快点
72 bign operator * (const int num) {
73     bign ret;
74     ret.len = 0;
75     int bb = 0;
76     for(int i = 0; i < len; i++) {

```



```

77         int now = bb + s[i]*num;
78         ret.s[ret.len++] = now%10;
79         bb = now/10;
80     }
81     while(bb) {
82         ret.s[ret.len++] = bb % 10;
83         bb /= 10;
84     }
85     ret.deal();
86     return ret;
87 }
88
89 bign operator / (const bign &a) const {
90     bign ret, cur = 0;
91     ret.len = len;
92     for(int i = len-1; i >= 0; i--) {
93         cur = cur*10;
94         cur.s[0] = s[i];
95         while(cur >= a) {
96             cur -= a;
97             ret.s[i]++;
98         }
99     }
100     ret.deal();
101     return ret;
102 }
103
104 bign operator % (const bign &a) const {
105     bign b = *this / a;
106     return *this - b*a;
107 }
108
109 bign operator += (const bign &a) { *this = *this + a; return *this; }
110 bign operator -= (const bign &a) { *this = *this - a; return *this; }
111 bign operator *= (const bign &a) { *this = *this * a; return *this; }
112 bign operator /= (const bign &a) { *this = *this / a; return *this; }
113 bign operator %= (const bign &a) { *this = *this % a; return *this; }
114
115 bool operator < (const bign &a) const {
116     if(len != a.len) return len < a.len;
117     for(int i = len-1; i >= 0; i--) if(s[i] != a.s[i])
118         return s[i] < a.s[i];
119     return false;
120 }
121 bool operator > (const bign &a) const { return a < *this; }
122 bool operator <= (const bign &a) const { return !(*this > a); }
123 bool operator >= (const bign &a) const { return !(*this < a); }
124 bool operator == (const bign &a) const { return !(*this > a || *this <
a); }

```

```

125     bool operator != (const bign &a) const { return *this > a || *this <
a; }
126
127     string str() const {
128         string ret = "";
129         for(int i = 0; i < len; i++) ret = char(s[i] + '0') + ret;
130         return ret;
131     }
132 };
133 istream& operator >> (istream &in, bign &x) {
134     string s;
135     in >> s;
136     x = s.c_str();
137     return in;
138 }
139 ostream& operator << (ostream &out, const bign &x) {
140     out << x.str();
141     return out;
142 }
143 // 大数开平方
144 bign Sqrt(bign x) {
145     int a[numlen/2];
146     int top = 0;
147     for(int i = 0; i < x.len; i += 2) {
148         if(i == x.len-1) {
149             a[top++] = x.s[i];
150         }
151         else
152             a[top++] = x.s[i] + x.s[i+1]*10;
153     }
154     bign ret = (int)sqrt((double)a[top-1]);
155     int xx = (int)sqrt((double)a[top-1]);
156     bign pre = a[top-1] - xx*xx;
157     bign cc;
158     for(int i = top-2; i >= 0; i--) {
159         pre = pre*100 + a[i];
160         cc = ret*20;
161         for(int j = 9; j >= 0; j--) {
162             bign now = (cc + j)*j;
163             if(now <= pre) {
164                 ret = ret*10 + j;
165                 pre -= now;
166                 break;
167             }
168         }
169     }
170     return ret;
171 }

```

```

1  /*
2      小根堆_笛卡尔树 O(n)
3      1. 结点——对应于数列元素。即数列中的每个元素都对应于树中某个唯一结点，树结点也对应
        于数列中的某个唯一元素
4      2. 中序遍历 (in-order traverse) 笛卡尔树即可得到原数列。即任意树结点的左子树结点
        所对应的数列元素下标比该结点所对应元素的下标小，右子树结点所对应数列元素下标比该结点所对应
        元素下标大。
5      3. 树结构存在堆序性质，即任意树结点所对应数值大 / 小于其左、右子树内任意结点对应数值
6  */
7
8  int stk[N];
9
10 for (int i = 1, top = 0, k; i <= n; i++) {
11     k = top;
12     while (k > 0 && p[stk[k]] > p[i]) k--;
13     if (k) rs[stk[k]] = i;
14     if (k < top) ls[i] = stk[k + 1];
15     stk[++k] = i;
16     top = k;
17 }

```

```

1  /*
2      Overflow?
3  */
4  ll pre[N][N];
5
6  void pre(int n, int m) {
7      for (int i = 1; i <= n; i++)
8          for (int j = 1; j <= m; j++)
9              pre[i][j] = pre[i - 1][j] + pre[i][j - 1] - pre[i - 1][j - 1] +
10 a[i][j];
11 }
12
13 ll calc(int x, int y, int u, int v) {
14     return pre[u][v] + pre[x - 1][y - 1] - pre[x - 1][v] - pre[u][y - 1];
15 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)

```

```

9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e6 + 5;
24
25 int n;
26 double a[N][N];
27
28 void Gauss_Jordan() {
29     for (int i = 1; i <= n; i++) {
30         int mx = i;
31         for (int j = i + 1; j <= n; j++) if (fabs(a[j][i]) > fabs(a[mx]
32 [i])) mx = j;
33         if (i != mx) for (int j = 1; j <= n + 1; j++) swap(a[i][j], a[mx]
34 [j]);
35         if (!a[i][i]) {
36             cout << "No Solution\n";
37             exit(0);
38         }
39         for (int j = 1; j <= n; j++) {
40             if (j != i) {
41                 double tmp = a[j][i] / a[i][i];
42                 for (int k = i + 1; k <= n + 1; k++) a[j][k] -= a[i][k] *
43 tmp;
44             }
45         }
46     }
47     for (int i = 1; i <= n; i++) a[i][n + 1] = a[i][n + 1] / a[i][i];
48 }
49
50 int main() {
51     ios::sync_with_stdio(0);
52     cin.tie(0);
53     cin >> n;
54     for (int i = 1; i <= n; i++) {
55         for (int j = 1; j <= n + 1; j++) {
56             cin >> a[i][j];
57         }
58     }
59 }

```

```

55     }
56     Gauss_Jordan();
57     for (int i = 1; i <= n; i++)
58         cout << fixed << setprecision(2) << a[i][n + 1] << '\n';
59     return 0;
60 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e5 + 5;
24
25 ll n, t[N], sum[N], len[N], sz[N], ans[N], T;
26 vector<pii> G[N];
27
28 void dfs(int u, int p) {
29     sum[u] = t[u];
30     sz[u] = 1;
31     for (auto [v, w] : G[u]) {
32         if (v == p) continue;
33         dfs(v, u);
34         sum[u] += sum[v];
35         sz[u] += sz[v];
36         len[1] += sz[v] * w;
37         ans[1] += (sum[v] + sz[v] * t[1]) * w;
38     }
39 }
40
41 void dp(int u, int p) {

```

```

42     for (auto [v, w] : G[u]) {
43         if (v == p) continue;
44         len[v] = len[u] - w * sz[v] + w * (n - sz[v]);
45         ans[v] = ans[u] - w * sum[v] + w * (T - sum[v]) - len[u] * t[u] +
len[v] * t[v];
46         dp(v, u);
47     }
48 }
49
50 int main() {
51     ios::sync_with_stdio(false);
52     cin.tie(nullptr);
53
54     cin >> n;
55     for (int i = 1; i <= n; i++) cin >> t[i];
56     T = accumulate(t + 1, t + 1 + n, 0);
57     for (int i = 1; i < n; i++) {
58         int u, v, w;
59         cin >> u >> v >> w;
60         G[u].push_back({v, w});
61         G[v].push_back({u, w});
62     }
63     dfs(1, 0);
64     dp(1, 0);
65     for (int i = 1; i <= n; i++) cout << ans[i] << '\n';
66
67     return 0;
68 }

```

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  #define rep(i, a, n) for (int i=a;i<n;i++)
5  #define per(i, a, n) for (int i=n-1;i>=a;i--)
6  #define pb push_back
7  #define mp make_pair
8  #define all(x) (x).begin(),(x).end()
9  #define fi first
10 #define se second
11 #define SZ(x) ((int)(x).size())
12 typedef vector<int> VI;
13 typedef long long ll;
14 typedef pair<int, int> PII;
15 typedef double db;
16 mt19937 mrand(random_device{}());
17 const ll mod = 1000000007;
18
19 int rnd(int x) { return mrand() % x; }

```

```

20
21 ll powmod(ll a, ll b) {
22     ll res = 1;
23     a %= mod;
24     assert(b >= 0);
25     for (; b; b >>= 1) {
26         if (b & 1) res = res * a % mod;
27         a = a * a % mod;
28     }
29     return res;
30 }
31
32 ll gcd(ll a, ll b) { return b ? gcd(b, a % b) : a; }
33 // head
34
35 typedef double db;
36 const db EPS = 1e-9;
37
38 inline int sign(db a) { return a < -EPS ? -1 : a > EPS; }
39
40 inline int cmp(db a, db b) { return sign(a - b); }
41
42 struct P {
43     db x, y;
44
45     P() {}
46
47     P(db _x, db _y) : x(_x), y(_y) {}
48
49     P operator+(P p) { return {x + p.x, y + p.y}; }
50
51     P operator-(P p) { return {x - p.x, y - p.y}; }
52
53     P operator*(db d) { return {x * d, y * d}; }
54
55     P operator/(db d) { return {x / d, y / d}; }
56
57     bool operator<(P p) const {
58         int c = cmp(x, p.x);
59         if (c) return c == -1;
60         return cmp(y, p.y) == -1;
61     }
62
63     bool operator==(P o) const {
64         return cmp(x, o.x) == 0 && cmp(y, o.y) == 0;
65     }
66
67     db dot(P p) { return x * p.x + y * p.y; }
68

```

```

69     db det(P p) { return x * p.y - y * p.x; }
70
71     db distTo(P p) { return (*this - p).abs(); }
72
73     db alpha() { return atan2(y, x); }
74
75     void read() { cin >> x >> y; }
76
77     void write() { cout << "(" << x << "," << y << ")" << endl; }
78
79     db abs() { return sqrt(abs2()); }
80
81     db abs2() { return x * x + y * y; }
82
83     P rot90() { return P(-y, x); }
84
85     P unit() { return *this / abs(); }
86
87     int quad() const { return sign(y) == 1 || (sign(y) == 0 && sign(x) >=
0); }
88
89     P rot(db an) { return {x * cos(an) - y * sin(an), x * sin(an) + y *
cos(an)}; }
90 };
91
92 struct L { //ps[0] -> ps[1]
93     P ps[2];
94
95     P &operator[](int i) { return ps[i]; }
96
97     P dir() { return ps[1] - ps[0]; }
98
99     L(P a, P b) {
100         ps[0] = a;
101         ps[1] = b;
102     }
103
104     bool include(P p) { return sign((ps[1] - ps[0]).det(p - ps[0])) > 0; }
105
106     L push() { // push eps outward
107         const double eps = 1e-8;
108         P delta = (ps[1] - ps[0]).rot90().unit() * eps;
109         return {ps[0] + delta, ps[1] + delta};
110     }
111 };
112
113 #define cross(p1, p2, p3) ((p2.x-p1.x)*(p3.y-p1.y)-(p3.x-p1.x)*(p2.y-
p1.y))
114 #define crossOp(p1, p2, p3) sign(cross(p1,p2,p3))

```



```

115
116 bool chkLL(P p1, P p2, P q1, P q2) {
117     db a1 = cross(q1, q2, p1), a2 = -cross(q1, q2, p2);
118     return sign(a1 + a2) != 0;
119 }
120
121 P isLL(P p1, P p2, P q1, P q2) {
122     db a1 = cross(q1, q2, p1), a2 = -cross(q1, q2, p2);
123     return (p1 * a2 + p2 * a1) / (a1 + a2);
124 }
125
126 P isLL(L l1, L l2) { return isLL(l1[0], l1[1], l2[0], l2[1]); }
127
128 bool intersect(db l1, db r1, db l2, db r2) {
129     if (l1 > r1) swap(l1, r1);
130     if (l2 > r2) swap(l2, r2);
131     return !(cmp(r1, l2) == -1 || cmp(r2, l1) == -1);
132 }
133
134 bool isSS(P p1, P p2, P q1, P q2) {
135     return intersect(p1.x, p2.x, q1.x, q2.x) && intersect(p1.y, p2.y,
136     q1.y, q2.y) &&
137     crossOp(p1, p2, q1) * crossOp(p1, p2, q2) <= 0 && crossOp(q1,
138     q2, p1)
139     * crossOp(q1,
140     q2, p2) <= 0;
141 }
142
143 bool isSS_strict(P p1, P p2, P q1, P q2) {
144     return crossOp(p1, p2, q1) * crossOp(p1, p2, q2) < 0 && crossOp(q1,
145     q2, p1)
146     * crossOp(q1,
147     q2, p2) < 0;
148 }
149
150 bool isMiddle(db a, db m, db b) {
151     return sign(a - m) == 0 || sign(b - m) == 0 || (a < m != b < m);
152 }
153
154 bool isMiddle(P a, P m, P b) {
155     return isMiddle(a.x, m.x, b.x) && isMiddle(a.y, m.y, b.y);
156 }
157
158 bool onSeg(P p1, P p2, P q) {
159     return crossOp(p1, p2, q) == 0 && isMiddle(p1, q, p2);
160 }
161
162 bool onSeg_strict(P p1, P p2, P q) {

```

```

158     return crossOp(p1, p2, q) == 0 && sign((q - p1).dot(p1 - p2)) *
159     sign((q - p2).dot(p1 - p2)) < 0;
160 }
161 P proj(P p1, P p2, P q) {
162     P dir = p2 - p1;
163     return p1 + dir * (dir.dot(q - p1) / dir.abs2());
164 }
165
166 P reflect(P p1, P p2, P q) {
167     return proj(p1, p2, q) * 2 - q;
168 }
169
170 db nearest(P p1, P p2, P q) {
171     P h = proj(p1, p2, q);
172     if (isMiddle(p1, h, p2))
173         return q.distTo(h);
174     return min(p1.distTo(q), p2.distTo(q));
175 }
176
177 db disSS(P p1, P p2, P q1, P q2) {
178     if (isSS(p1, p2, q1, q2)) return 0;
179     return min(min(nearest(p1, p2, q1), nearest(p1, p2, q2)),
180 min(nearest(q1, q2, p1), nearest(q1, q2, p2)));
181 }
182
183 db rad(P p1, P p2) {
184     return atan2l(p1.det(p2), p1.dot(p2));
185 }
186
187 db incircle(P p1, P p2, P p3) {
188     db A = p1.distTo(p2);
189     db B = p2.distTo(p3);
190     db C = p3.distTo(p1);
191     return sqrtl(A * B * C / (A + B + C));
192 }
193 //polygon
194
195 db area(vector<P> ps) {
196     db ret = 0;
197     rep(i, 0, ps.size()) ret += ps[i].det(ps[(i + 1) % ps.size()]);
198     return ret / 2;
199 }
200
201 int contain(vector<P> ps, P p) { //2:inside,1:on_seg,0:outside
202     int n = ps.size(), ret = 0;
203     rep(i, 0, n) {
204         P u = ps[i], v = ps[(i + 1) % n];

```

```

205         if (onSeg(u, v, p)) return 1;
206         if (cmp(u.y, v.y) <= 0) swap(u, v);
207         if (cmp(p.y, u.y) > 0 || cmp(p.y, v.y) <= 0) continue;
208         ret ^= crossOp(p, u, v) > 0;
209     }
210     return ret * 2;
211 }
212
213 vector<P> convexHull(vector<P> ps) {
214     int n = ps.size();
215     if (n <= 1) return ps;
216     sort(ps.begin(), ps.end());
217     vector<P> qs(n * 2);
218     int k = 0;
219     for (int i = 0; i < n; qs[k++] = ps[i++])
220         while (k > 1 && crossOp(qs[k - 2], qs[k - 1], ps[i]) <= 0) --k;
221     for (int i = n - 2, t = k; i >= 0; qs[k++] = ps[i--])
222         while (k > t && crossOp(qs[k - 2], qs[k - 1], ps[i]) <= 0) --k;
223     qs.resize(k - 1);
224     return qs;
225 }
226
227 vector<P> convexHullNonStrict(vector<P> ps) {
228     //caution: need to unique the Ps first
229     int n = ps.size();
230     if (n <= 1) return ps;
231     sort(ps.begin(), ps.end());
232     vector<P> qs(n * 2);
233     int k = 0;
234     for (int i = 0; i < n; qs[k++] = ps[i++])
235         while (k > 1 && crossOp(qs[k - 2], qs[k - 1], ps[i]) < 0) --k;
236     for (int i = n - 2, t = k; i >= 0; qs[k++] = ps[i--])
237         while (k > t && crossOp(qs[k - 2], qs[k - 1], ps[i]) < 0) --k;
238     qs.resize(k - 1);
239     return qs;
240 }
241
242 db convexDiameter(vector<P> ps) {
243     int n = ps.size();
244     if (n <= 1) return 0;
245     int is = 0, js = 0;
246     rep(k, 1, n) is = ps[k] < ps[is] ? k : is, js = ps[js] < ps[k] ? k :
js;
247     int i = is, j = js;
248     db ret = ps[i].distTo(ps[j]);
249     do {
250         if ((ps[(i + 1) % n] - ps[i]).det(ps[(j + 1) % n] - ps[j]) >= 0)
251             (++j) %= n;
252         else

```

```

253         (++i) %= n;
254         ret = max(ret, ps[i].distTo(ps[j]));
255     } while (i != is || j != js);
256     return ret;
257 }
258
259 vector<P> convexCut(const vector<P> &ps, P q1, P q2) {
260     vector<P> qs;
261     int n = ps.size();
262     rep(i, 0, n) {
263         P p1 = ps[i], p2 = ps[(i + 1) % n];
264         int d1 = crossOp(q1, q2, p1), d2 = crossOp(q1, q2, p2);
265         if (d1 >= 0) qs.pb(p1);
266         if (d1 * d2 < 0) qs.pb(isLL(p1, p2, q1, q2));
267     }
268     return qs;
269 }
270
271 //min_dist
272
273 db min_dist(vector<P> &ps, int l, int r) {
274     if (r - l <= 5) {
275         db ret = 1e100;
276         rep(i, l, r) rep(j, l, i) ret = min(ret, ps[i].distTo(ps[j]));
277         return ret;
278     }
279     int m = (l + r) >> 1;
280     db ret = min(min_dist(ps, l, m), min_dist(ps, m, r));
281     vector<P> qs;
282     rep(i, l, r) if (abs(ps[i].x - ps[m].x) <= ret) qs.pb(ps[i]);
283     sort(qs.begin(), qs.end(), [](P a, P b) -> bool { return a.y < b.y;
284 });
285     rep(i, 1, qs.size()) for (int j = i - 1; j >= 0 && qs[j].y >= qs[i].y
286 - ret; --j)
287         ret = min(ret, qs[i].distTo(qs[j]));
288     return ret;
289 }
290
291 int type(P o1, db r1, P o2, db r2) {
292     db d = o1.distTo(o2);
293     if (cmp(d, r1 + r2) == 1) return 4;
294     if (cmp(d, r1 + r2) == 0) return 3;
295     if (cmp(d, abs(r1 - r2)) == 1) return 2;
296     if (cmp(d, abs(r1 - r2)) == 0) return 1;
297     return 0;
298 }
299
300 vector<P> isCL(P o, db r, P p1, P p2) {
301     if (cmp(abs((o - p1).det(p2 - p1) / p1.distTo(p2)), r) > 0) return {};

```

```

300     db x = (p1 - o).dot(p2 - p1), y = (p2 - p1).abs2(), d = x * x - y *
((p1 - o).abs2() - r * r);
301     d = max(d, 0.0);
302     P m = p1 - (p2 - p1) * (x / y), dr = (p2 - p1) * (sqrt(d) / y);
303     return {m - dr, m + dr}; //along dir: p1->p2
304 }
305
306 vector<P> isCC(P o1, db r1, P o2, db r2) { //need to check whether two
circles are the same
307     db d = o1.distTo(o2);
308     if (cmp(d, r1 + r2) == 1) return {};
309     if (cmp(d, abs(r1 - r2)) == -1) return {};
310     d = min(d, r1 + r2);
311     db y = (r1 * r1 + d * d - r2 * r2) / (2 * d), x = sqrt(r1 * r1 - y *
y);
312     P dr = (o2 - o1).unit();
313     P q1 = o1 + dr * y, q2 = dr.rot90() * x;
314     return {q1 - q2, q1 + q2}; //along circle 1
315 }
316
317 vector<P> tanCP(P o, db r, P p) {
318     db x = (p - o).abs2(), d = x - r * r;
319     if (sign(d) <= 0) return {}; // on circle => no tangent
320     P q1 = o + (p - o) * (r * r / x);
321     P q2 = (p - o).rot90() * (r * sqrt(d) / x);
322     return {q1 - q2, q1 + q2}; //counter clock-wise
323 }
324
325
326 vector<L> extanCC(P o1, db r1, P o2, db r2) {
327     vector<L> ret;
328     if (cmp(r1, r2) == 0) {
329         P dr = (o2 - o1).unit().rot90() * r1;
330         ret.pb(L(o1 + dr, o2 + dr)), ret.pb(L(o1 - dr, o2 - dr));
331     } else {
332         P p = (o2 * r1 - o1 * r2) / (r1 - r2);
333         vector<P> ps = tanCP(o1, r1, p), qs = tanCP(o2, r2, p);
334         rep(i, 0, min(ps.size(), qs.size())) ret.pb(L(ps[i], qs[i])); //c1
counter-clock wise
335     }
336     return ret;
337 }
338
339 vector<L> intanCC(P o1, db r1, P o2, db r2) {
340     vector<L> ret;
341     P p = (o1 * r2 + o2 * r1) / (r1 + r2);
342     vector<P> ps = tanCP(o1, r1, p), qs = tanCP(o2, r2, p);
343     rep(i, 0, min(ps.size(), qs.size())) ret.pb(L(ps[i], qs[i])); //c1
counter-clock wise

```

```

344     return ret;
345 }
346
347 db areaCT(db r, P p1, P p2) {
348     vector<P> is = isCL(P(0, 0), r, p1, p2);
349     if (is.empty()) return r * r * rad(p1, p2) / 2;
350     bool b1 = cmp(p1.abs2(), r * r) == 1, b2 = cmp(p2.abs2(), r * r) == 1;
351     if (b1 && b2) {
352         if (sign((p1 - is[0]).dot(p2 - is[0])) <= 0 &&
353             sign((p1 - is[0]).dot(p2 - is[0])) <= 0)
354             return r * r * (rad(p1, is[0]) + rad(is[1], p2)) / 2 +
355             is[0].det(is[1]) / 2;
356         else return r * r * rad(p1, p2) / 2;
357     }
358     if (b1) return (r * r * rad(p1, is[0]) + is[0].det(p2)) / 2;
359     if (b2) return (p1.det(is[1]) + r * r * rad(is[1], p2)) / 2;
360     return p1.det(p2) / 2;
361 }
362
363 bool parallel(L l0, L l1) { return sign(l0.dir().det(l1.dir())) == 0; }
364
365 bool sameDir(L l0, L l1) { return parallel(l0, l1) &&
366     sign(l0.dir().dot(l1.dir())) == 1; }
367
368 bool cmp(P a, P b) {
369     if (a.quad() != b.quad()) {
370         return a.quad() < b.quad();
371     } else {
372         return sign(a.det(b)) > 0;
373     }
374 }
375
376 bool operator<(L l0, L l1) {
377     if (sameDir(l0, l1)) {
378         return l1.include(l0[0]);
379     } else {
380         return cmp(l0.dir(), l1.dir());
381     }
382 }
383
384 bool check(L u, L v, L w) {
385     return w.include(isLL(u, v));
386 }
387
388 vector<P> halfPlaneIS(vector<L> &l) {
389     sort(l.begin(), l.end());
390     deque<L> q;
391     for (int i = 0; i < (int) l.size(); ++i) {
392         if (i && sameDir(l[i], l[i - 1])) continue;

```

```

391         while (q.size() > 1 && !check(q[q.size() - 2], q[q.size() - 1],
1[i])) q.pop_back();
392         while (q.size() > 1 && !check(q[1], q[0], l[i])) q.pop_front();
393         q.push_back(l[i]);
394     }
395     while (q.size() > 2 && !check(q[q.size() - 2], q[q.size() - 1], q[0]))
q.pop_back();
396     while (q.size() > 2 && !check(q[1], q[0], q[q.size() - 1]))
q.pop_front();
397     vector<P> ret;
398     for (int i = 0; i < (int) q.size(); ++i) ret.push_back(isLL(q[i], q[(i
+ 1) % q.size()]));
399     return ret;
400 }
401
402 P inCenter(P A, P B, P C) {
403     double a = (B - C).abs(), b = (C - A).abs(), c = (A - B).abs();
404     return (A * a + B * b + C * c) / (a + b + c);
405 }
406
407 P circumCenter(P a, P b, P c) {
408     P bb = b - a, cc = c - a;
409     double db = bb.abs2(), dc = cc.abs2(), d = 2 * bb.det(cc);
410     return a - P(bb.y * dc - cc.y * db, cc.x * db - bb.x * dc) / d;
411 }
412
413 P othroCenter(P a, P b, P c) {
414     P ba = b - a, ca = c - a, bc = b - c;
415     double Y = ba.y * ca.y * bc.y,
416            A = ca.x * ba.y - ba.x * ca.y,
417            x0 = (Y + ca.x * ba.y * b.x - ba.x * ca.y * c.x) / A,
418            y0 = -ba.x * (x0 - c.x) / ba.y + ca.y;
419     return {x0, y0};
420 }
421
422 const int N = 201000;
423 int n, H;
424 int x[N], y[N];
425 P p[N];
426 db ans;
427
428 int main() {
429     scanf("%d%d", &n, &H);
430     rep(i, 1, n + 1) {
431         scanf("%d%d", x + i, y + i);
432         p[i] = P(x[i], y[i]);
433     }
434     P eye(x[n], y[n] + H);
435     P block(x[n], y[n]);

```

```

436     per(i, 1, n) {
437         bool canseeR = crossOp(eye, block, p[i + 1]) <= 0;
438         bool canseeL = crossOp(eye, block, p[i]) <= 0;
439         if (canseeL) {
440             if (canseeR) ans += p[i].distTo(p[i + 1]);
441             else {
442                 P inter = isLL(eye, block, p[i], p[i + 1]);
443                 ans += p[i].distTo(inter);
444             }
445         }
446         if (canseeL) block = p[i];
447     }
448     printf("%.10f\n", ans);
449 }

```

```

1  inline ll qpow(ll a, ll b, ll p = mod){
2      ll ans = 1;
3      a %= p;
4      while(b>0){
5          if(b & 1) ans = (ans * a) % p;
6          a = (a * a) % p;
7          b >>= 1;
8      }
9      return ans;
10 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 1e6 + 5;
15
16 int n, q, a[N], x, y, l = 1, r, p[N], ans[N], sq, belong[N], num;
17
18 struct node {
19     int l, r, bel;
20
21     bool operator<(const node &T) const {

```



```

22         return (belong[l] ^ belong[T.l]) ? belong[l] < belong[T.l] :
((belong[l] & 1) ? r < T.r : r > T.r);
23     }
24 } query[N];
25
26 int cmp(node a, node b) {
27     if (a.l / sq != b.l / sq) return a.l / sq < b.l / sq;
28     return a.r / sq < b.r / sq;
29 }
30
31 int main() {
32     ios::sync_with_stdio(false);
33     cin.tie(nullptr);
34     cin >> n >> q;
35     for (int i = 1; i <= n; i++) cin >> a[i];
36     for (int i = 1; i <= q; i++) cin >> x >> y, query[i] = {x, y, i};
37     sq = sqrt(n);
38     for (int i = 1; i <= n; i++) belong[i] = min(sq, (i - 1) / sq + 1);
39     sort(query + 1, query + 1 + q);
40     for (int i = 1; i <= q; i++) {
41         while (query[i].l < 1) num += !p[a[--l]]++;
42         while (query[i].l > 1) num -= !--p[a[l++]];
43         while (query[i].r < r) num -= !--p[a[r--]];
44         while (query[i].r > r) num += !p[a[++r]]++;
45         ans[query[i].bel] = num;
46     }
47     for (int i = 1; i <= q; i++) cout << ans[i] << '\n';
48     return 0;
49 }

```

```

1 using db = double;
2
3 pair<db, db> solve(db X1, db Y1, db X2, db Y2, db X3, db Y3) {
4     double X = ((Y2 - Y1) * (Y3 * Y3 - Y1 * Y1 + X3 * X3 - X1 * X1) - (Y3 -
Y1) * (Y2 * Y2 - Y1 * Y1 + X2 * X2 - X1 * X1)) / (2.0 * ((X3 - X1) * (Y2 -
Y1) - (X2 - X1) * (Y3 - Y1)));
5     double Y = ((X2 - X1) * (X3 * X3 - X1 * X1 + Y3 * Y3 - Y1 * Y1) - (X3 -
X1) * (X2 * X2 - X1 * X1 + Y2 * Y2 - Y1 * Y1)) / (2.0 * ((Y3 - Y1) * (X2 -
X1) - (Y2 - Y1) * (X3 - X1)));
6     return make_pair(X, Y);
7 }

```

```

1 #include <bits/stdc++.h>
2
3 #define fi first
4 #define se second
5 #define mp make_pair
6 #define pb push_back

```

```

7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e6 + 5;
24
25 int n, a, b, da, db;
26 int diam, d[N], f[N];
27 vector<int> G[N];
28
29 void dfs(int u, int p) {
30     f[u] = 0;
31     for (auto v:G[u]) {
32         if (v != p) {
33             d[v] = d[u] + 1;
34             dfs(v, u);
35             diam = max(diam, f[u] + f[v] + 1);
36             f[u] = max(f[u], f[v] + 1);
37         }
38     }
39 }
40
41 void solve() {
42     cin >> n >> a >> b >> da >> db;
43     for (int i = 1; i <= n; i++) G[i].clear();
44     for (int i = 1; i < n; i++) {
45         int u, v;
46         cin >> u >> v;
47         G[u].emplace_back(v);
48         G[v].emplace_back(u);
49     }
50     diam = d[a] = 0;
51     dfs(a, -1);
52     cout << (2 * da >= min(diam, db) || d[b] <= da ? "Alice" : "Bob") <<
53     '\n';
54 }

```

```

55 int main() {
56     ios::sync_with_stdio(false);
57     cin.tie(nullptr);
58     int T;
59     cin >> T;
60     while (T--) solve();
61     return 0;
62 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e6 + 5;
24
25 int n, a, b, da, db;
26 int diam, d[N];
27 vector<int> G[N];
28
29 int dfs(int u, int p) {
30     int len = 0;
31     for (auto v:G[u]) {
32         if (v != p) {
33             d[v] = d[u] + 1;
34             int cur = dfs(v, u) + 1;
35             diam = max(diam, cur + len);
36             len = max(len, cur);
37         }
38     }
39     return len;

```

```

40 }
41
42 void solve() {
43     cin >> n >> a >> b >> da >> db;
44     for (int i = 1; i <= n; i++) G[i].clear();
45     for (int i = 1; i < n; i++) {
46         int u, v;
47         cin >> u >> v;
48         G[u].emplace_back(v);
49         G[v].emplace_back(u);
50     }
51     diam = d[a] = 0;
52     dfs(a, -1);
53     cout << (2 * da >= min(diam, db) || d[b] <= da ? "Alice" : "Bob") <<
54     '\n';
55 }
56
57 int main() {
58     ios::sync_with_stdio(false);
59     cin.tie(nullptr);
60     int T;
61     cin >> T;
62     while (T--) solve();
63     return 0;
64 }

```

```

1  template <typename T> struct fenwick_tree {
2      int n;
3      vector<T> t;
4      fenwick_tree() : n(0) {}
5      fenwick_tree(int n) : n(n), t(n + 1) {}
6
7      void add(int p, T x) {
8          while (p <= n) {
9              t[p] += x;
10             p += lowbit(p);
11         }
12     }
13
14     T sum(int x) {
15         T s = 0;
16         while (x > 0) {
17             s += t[x];
18             x -= lowbit(x);
19         }
20         return s;
21     }
22 }

```

```

23     T sum(int l, int r) {
24         return sum(r) - sum(l - 1);
25     }
26 };

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e6 + 5;
24
25 string s;
26 int f[N][17], n, K;
27
28 int main() {
29     ios::sync_with_stdio(false);
30     cin.tie(nullptr);
31
32     cin >> s >> K;
33     n = s.size();
34     for (auto &c : s) {
35         if (isdigit(c)) c -= '0';
36         else c -= 'A' - 10;
37     }
38     int mask = 0;
39     for (int i = 0; i < n; i++) {
40         for (int j = 1; j <= 16; j++) {
41             f[i + 1][j] = (f[i + 1][j] + 1ll * f[i][j] * j % mod) % mod;
42             if (j < 16) f[i + 1][j + 1] = (f[i + 1][j + 1] + 1ll * f[i][j]
* (16 - j) % mod) % mod;

```

```

43     }
44     f[i + 1][1] = (f[i + 1][1] + 111 * f[i][0] * 15 % mod) % mod;
45     f[i + 1][0] = (f[i + 1][0] + f[i][0]) % mod;
46     for (int j = 0; j < s[i]; j++) {
47         int MASK = mask;
48         if (i || j) MASK |= 1 << j;
49         f[i + 1][__builtin_popcount(MASK)]++;
50     }
51     mask |= 1 << s[i];
52 }
53 int ans = (f[n][K] + (__builtin_popcount(mask) == K)) % mod;
54 cout << ans << '\n';
55
56 return 0;
57 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define lowbit(x) ((x) & -(x))
7
8  using namespace std;
9  using ll = long long;
10 using pii = pair<int, int>;
11
12 constexpr double eps = 1e-8;
13 constexpr int NINF = 0xc0c0c0c0;
14 constexpr int INF = 0x3f3f3f3f;
15 constexpr ll mod = 1e9 + 7;
16 constexpr ll N = 1e6 + 5;
17
18 ll f[15][15][2][2], digit;
19 string s;
20
21 ll dfs(int pos, int sum, bool limit, bool zero) {
22     ll ans = 0;
23     if (pos == s.size()) return sum;
24     if (f[pos][sum][limit][zero] != -1) return f[pos][sum][limit][zero];
25     for (int i = 0; i <= (limit ? s[pos] - '0' : 9); i++) {
26         if (zero && i == 0) {
27             ans += dfs(pos + 1, sum, limit && i == s[pos] - '0', true);
28         } else {
29             ans += dfs(pos + 1, sum + (i == digit), limit && i == s[pos] -
30                 '0', false);
31         }
32     }
33 }

```

```

32     return f[pos][sum][limit][zero] = ans;
33 }
34
35 ll calc(ll x) {
36     memset(f, -1, sizeof(f));
37     s = to_string(x);
38     return dfs(0, 0, true, true);
39 }
40
41 int main() {
42     ios::sync_with_stdio(false);
43     cin.tie(nullptr);
44
45     ll l, r;
46     cin >> l >> r;
47     for (int i = 0; i <= 9; i++) {
48         digit = i;
49         cout << calc(r) - calc(l - 1) << " \n"[i == 9];
50     }
51
52     return 0;
53 }

```

```

1  #include<bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  using namespace std;
7  typedef long long ll;
8  typedef pair<int, int> P;
9  const double eps = 1e-8;
10 const int NINF = 0xc0c0c0c0;
11 const int INF = 0x3f3f3f3f;
12 const ll mod = 1e9 + 7;
13 const ll N = 1e5 + 5;
14
15 int n, m;
16 vector<int> G[N];
17 int a[N], cnt, deep[N], deg[N];
18
19 void topsort() {
20     queue<int> q;
21     for (int i = 1; i <= n; i++) if (!deg[i]) q.push(i), deep[i] = 1;
22     while (q.size()) {
23         int p = q.front();
24         q.pop();
25         a[++cnt] = p;

```

```

26         for (auto c:G[p]) {
27             deep[c] = deep[p] + 1;
28             if (!--deg[c]) q.push(c);
29         }
30     }
31 }
32
33 int main() {
34     ios::sync_with_stdio(false);
35     cin.tie(0);
36     cin >> n >> m;
37     for (int i = 1; i <= m; i++) {
38         int u, v;
39         cin >> u >> v;
40         G[u].push_back(v);
41         ++deg[v];
42     }
43     topsort();
44     for (int i = 1; i <= n; i++) {
45         cout << deep[i] << '\n';
46     }
47     return 0;
48 }

```

```

1  int deg[N];
2  vector<int> top_idx;
3
4  bool topsort(int n) {
5      queue<int> q;
6      for (int i = 1; i <= n; i++) if (!deg[i]) q.push(i);
7      while (!q.empty()) {
8          int u = q.front();
9          q.pop();
10         top_idx.push_back(u);
11         for (auto v : G[u]) if (!--deg[v]) q.push(v);
12     }
13     return top_idx.size() == n;
14 }

```

```

1  template<typename T>
2  T exgcd(T a, T b, T &x, T &y) {
3      if (!b) { x = 1, y = 0; return a; }
4      T d = exgcd(b, a % b, x, y);
5      T z = x;
6      x = y, y = z - y * (a / b);
7      return d;
8  }
9

```



```

10 template<typename T>
11 T inv(T a, T p) {
12     T x, y;
13     exgcd(a, p, x, y);
14     return (x % p + p) % p;
15 }
16
17 /*
18     线性求逆元
19 */
20
21 vector<int> inverse(int n, int p) {
22     vector<int> inv(n + 1);
23     inv[1] = 1;
24     for (int i = 2; i <= n; i++) {
25         inv[i] = 1ll * (p - p / i) * inv[p % i] % p;
26     }
27     return inv;
28 }

```

```

1  ll fac[N + 5], ifac[N + 5];
2
3  ll qpow(ll a, ll b) {
4      ll res = 1;
5      while (b > 0) {
6          if (b & 1) res = res * a % mod;
7          a = a * a % mod;
8          b >>= 1;
9      }
10     return res;
11 }
12
13 inline ll perm(ll x, ll y) { return y > x || y < 0 ? 0 : fac[x] * ifac[x - y] % mod; }
14
15 inline ll comb(ll x, ll y) { return y > x || y < 0 ? 0 : perm(x, y) * ifac[y] % mod; }
16
17 void init() {
18     fac[0] = 1;
19     for (int i = 1; i <= N; i++) fac[i] = 1ll * i * fac[i - 1] % mod;
20     ifac[N] = qpow(fac[N], mod - 2);
21     for (int i = N; i; i--) ifac[i - 1] = 1ll * i * ifac[i] % mod;
22 }

```

```

1  #include<cstdio>
2  #include<cmath>
3  #include<algorithm>

```

```

4
5 using namespace std;
6 const double eps = 1e-3;
7
8 int n;
9
10 struct Point {
11     double x;
12     double y;
13     double z;
14
15     Point(double _x = 0.0, double _y = 0.0, double _z = 0.0) {
16         x = _x;
17         y = _y;
18         z = _z;
19     }
20 };
21
22 Point points[105];
23
24 inline double dis(Point a, Point b) {
25     double norm = (a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y) +
26     (a.z - b.z) * (a.z - b.z);
27     return sqrt(norm);
28 }
29
30 double solve() {
31     double T = 1000.0; //初始温度
32     double rate = 0.99999; //温度下降系数
33     Point ans_p; //初始圆心
34     int cur;
35     Point max_p; //距离圆心最远的点
36     double ans = 1e99;
37     while (T > eps) //模拟降温
38     {
39         double max_dis = 0.0; //选定圆心到最远点距离
40         for (int i = 1; i <= n; ++i) { //最远的点即points[cur]
41             if (dis(ans_p, points[i]) > max_dis) {
42                 max_dis = dis(ans_p, points[i]);
43                 cur = i;
44             }
45         }
46         ans = min(ans, max_dis);
47         ans_p.x += (points[cur].x - ans_p.x) * (T / 1000.0);
48         ans_p.y += (points[cur].y - ans_p.y) * (T / 1000.0);
49         ans_p.z += (points[cur].z - ans_p.z) * (T / 1000.0);
50         T *= rate;
51     }
52     return ans;

```

```

52 }
53
54 int main() {
55     scanf("%d", &n);
56     for (int i = 1; i <= n; ++i) {
57         scanf("%lf%lf%lf", &points[i].x, &points[i].y, &points[i].z);
58     }
59     printf("%.15f\n", solve());
60     return 0;
61 }

```

```

1  #include <bits/stdc++.h>
2  #include <algorithm>
3
4  using namespace std;
5  constexpr int N = 5000 + 5;
6
7  struct node {
8      double x, y, z;
9  } a[N];
10
11 struct vec {
12     double x, y;
13
14     vec(const double &x0 = 0, const double &y0 = 0) : x(x0), y(y0) {}
15
16     vec operator+(const vec &t) const { return vec(x + t.x, y + t.y); }
17
18     vec operator-(const vec &t) const { return vec(x - t.x, y - t.y); }
19
20     vec operator*(const double &t) const { return vec(x * t, y * t); }
21
22     vec operator/(const double &t) const { return vec(x / t, y / t); }
23
24     const double len2() const { return x * x + y * y; }
25
26     const double len() const { return sqrt(len2()); }
27
28     vec norm() const { return *this / len(); }
29
30     vec rotate_90_c() { return vec(y, -x); }
31 };
32
33 double dot(const vec &a, const vec &b) { return a.x * b.x + a.y * b.y; }
34
35 double crs(const vec &a, const vec &b) { return a.x * b.y - a.y * b.x; }
36

```

```

37 vec lin_lin_int(const vec &p0, const vec &v0, const vec &p1, const vec &v1)
38 {
39     double t = crs(p1 - p0, v1) / crs(v0, v1);
40     return p0 + v0 * t;
41 }
42
43 vec circle(const vec &a, const vec &b, const vec &c) {
44     return lin_lin_int((a + b) / 2, (b - a).rotate_90_c(), (a + c) / 2, (c
45 - a).rotate_90_c());
46 }
47
48 int n;
49 vec pot[100005];
50
51 double work() {
52     random_shuffle(pot + 1, pot + n + 1);
53     vec o;
54     double r2 = 0;
55     for (int i = 1; i <= n; i++) {
56         if ((pot[i] - o).len2() > r2) {
57             o = pot[i], r2 = 0;
58             for (int j = 1; j < i; j++) {
59                 if ((pot[j] - o).len2() > r2) {
60                     o = (pot[i] + pot[j]) / 2, r2 = (pot[j] - o).len2();
61                     for (int k = 1; k < j; k++) {
62                         if ((pot[k] - o).len2() > r2) {
63                             o = circle(pot[i], pot[j], pot[k]), r2 =
64 (pot[k] - o).len2();
65                         }
66                     }
67                 }
68             }
69         }
70     }
71     return 2.0 * sqrt(r2);
72 }
73
74 int main() {
75     scanf("%d", &n);
76     for (int i = 1; i <= n; i++) cin >> a[i].x >> a[i].y >> a[i].z;
77     for (int i = 1; i <= n; i++) {
78         pot[i].x = a[i].x;
79         pot[i].y = a[i].y;
80     }
81     double ans = work();
82     for (int i = 1; i <= n; i++) {
83         pot[i].x = a[i].x;
84         pot[i].y = a[i].z;
85     }

```

```

83     ans = min(ans, work());
84     for (int i = 1; i <= n; i++) {
85         pot[i].x = a[i].y;
86         pot[i].y = a[i].z;
87     }
88     ans = min(ans, work());
89     printf("%.10lf\n", ans);
90     return 0;
91 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 1e6 + 5;
15
16 int n, A[N];
17
18 int Min_show() {
19     int i = 0, j = 1, k = 0;
20     while (i < n and j < n and k < n) {
21         if (A[(i + k) % n] == A[(j + k) % n]) k++;
22         else {
23             A[(i + k) % n] > A[(j + k) % n] ? i += k + 1 : j += k + 1;
24             if (i == j) i++;
25             k = 0;
26         }
27     }
28     return min(i, j);
29 }
30
31 int Max_show() {
32     int i = 0, j = 1, k = 0;
33     while (i < n and j < n and k < n) {
34         if (A[(i + k) % n] == A[(j + k) % n]) k++;
35         else {
36             A[(i + k) % n] < A[(j + k) % n] ? i += k + 1 : j += k + 1;
37             if (i == j) i++;
38             k = 0;

```

```

39     }
40 }
41     return min(i, j);
42 }
43
44 int main() {
45     ios::sync_with_stdio(false);
46     cin.tie(nullptr);
47     cin >> n;
48     for (int i = 0; i < n; i++) {
49         cin >> A[i];
50     }
51     int s = Min_show();
52     for (int i = 0; i < n; i++) cout << A[(s + i) % n] << " \n"[i == n -
1];
53     return 0;
54 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 1e5 + 5;
15
16 int n, m;
17
18 struct BIT {
19     int bit[N], mi[N], ma[N];
20
21     void add(int x, int v) {
22         bit[x] = v;
23         for (; x <= n; x += x & -x) {
24             mi[x] = ma[x] = v;
25             for (int i = 1; i < (x & -x); i <= 1)
26                 mi[x] = min(mi[x], mi[x - i]),
27                 ma[x] = max(ma[x], ma[x - i]);
28         }
29     }
30 }

```

```

31     int difference(int l, int r) {
32         int L = bit[r], R = bit[r];
33         while (l <= r) {
34             L = min(L, bit[r]);
35             R = max(R, bit[r]);
36             for (--r; r - l >= (r & -r); r -= (r & -r))
37                 L = min(L, mi[r]),
38                 R = max(R, ma[r]);
39         }
40         return R - L + 1;
41     }
42 };
43
44 BIT a;
45
46 int main() {
47     ios::sync_with_stdio(false);
48     cin.tie(nullptr);
49     cin >> n >> m;
50     for (int i = 1; i <= n; i++) {
51         int x;
52         cin >> x;
53         a.add(i, x);
54     }
55     while (m--) {
56         int t, l, r;
57         cin >> t >> l >> r;
58         if (t == 1) a.add(l, r);
59         else cout << (a.difference(l, r) == r - l + 1 ? "YES" : "NO") <<
'\n';
60     }
61     return 0;
62 }

```

```

1  typedef long long ll;
2  #define sz(x) ((ll)x.size())
3  #define pb push_back
4  const ll mod = 1e9 + 7;
5
6  ll fpow(ll a, ll b) {
7      ll res = 1;
8      a %= mod;
9      assert(b >= 0);
10     for (; b; b >>= 1) {
11         if (b & 1) res = res * a % mod;
12         a = a * a % mod;
13     }
14     return res;

```

```

15 }
16
17 namespace linear_seq {
18     const ll maxn = 10010;
19     ll res[maxn], base[maxn], c[maxn], md[maxn];
20     vector<long long> v;
21
22     void mul(ll *a, ll *b, ll k) {
23         for (int i = 0; i < k + k; ++i) c[i] = 0;
24         for (int i = 0; i < k; ++i)
25             if (a[i])
26                 for (int j = 0; j < k; ++j)
27                     c[i + j] = (c[i + j] + a[i] * b[j]) % mod;
28         for (ll i = k + k - 1; i >= k; i--)
29             if (c[i])
30                 for (int j = 0; j < v.size(); ++j)
31                     c[i - k + v[j]] = (c[i - k + v[j]] - c[i] * md[v[j]])
% mod;
32         for (int i = 0; i < k; ++i) a[i] = c[i];
33     }
34
35     ll solve(ll n, vector <ll> a, vector <ll> b) {
36         ll ans = 0, cnt = 0, k = a.size();
37         for (int i = 0; i < k; ++i) md[k - 1 - i] = -a[i];
38         md[k] = 1;
39         v.clear();
40         for (int i = 0; i < k; ++i) if (md[i] != 0) v.push_back(i);
41         for (int i = 0; i < k; ++i) res[i] = base[i] = 0;
42         res[0] = 1;
43         while ((1ll << cnt) <= n) cnt++;
44         for (ll p = cnt; p >= 0; p--) {
45             mul(res, res, k);
46             if ((n >> p) & 1) {
47                 for (ll i = k - 1; i >= 0; --i) res[i + 1] = res[i];
48                 res[0] = 0;
49                 for (int j = 0; j < v.size(); ++j) res[v[j]] = (res[v[j]]
- res[k] * md[v[j]]) % mod;
50             }
51         }
52         for (int i = 0; i < k; ++i) ans = (ans + res[i] * b[i]) % mod;
53         if (ans < 0) ans += mod;
54         return ans;
55     }
56
57     vector <ll> bm(vector <ll> s) {
58         vector <ll> C(1, 1), B(1, 1);
59         ll k = 0, m = 1, b = 1;
60         for (int n = 0; n < s.size(); ++n) {
61             ll d = 0;

```



```

62         for (int i = 0; i <= k; ++i) d = (d + (ll) C[i] * s[n - i]) %
mod;
63         if (!d) ++m;
64         else if (k + k <= n) {
65             vector<ll> T = C;
66             ll c = mod - d * fpow(b, mod - 2) % mod;
67             while (sz(C) < sz(B) + m) C.pb(0);
68             for (int i = 0; i < B.size(); ++i) C[i + m] = (C[i + m] +
c * B[i]) % mod;
69             k = n + 1 - k;
70             B = T;
71             b = d;
72             m = 1;
73         } else {
74             ll c = mod - d * fpow(b, mod - 2) % mod;
75             while (sz(C) < sz(B) + m) C.pb(0);
76             for (int i = 0; i < B.size(); ++i) C[i + m] = (C[i + m] +
c * B[i]) % mod;
77             ++m;
78         }
79     }
80     return C;
81 }
82
83 ll gao(vector<ll> a, ll n) {
84     vector<ll> c = bm(a);
85     c.erase(c.begin());
86     for (int i = 0; i < c.size(); ++i) c[i] = (mod - c[i]) % mod;
87     return solve(n, c, vector<ll>(a.begin(), a.begin() + sz(c)));
88 }
89 };
90
91 class Solution {
92 public:
93     /**
94      * 返回c[n]%1000000007的值
95      * @param n long长整型 即题目中的n
96      * @return int整型
97      */
98     int Answerforcn(long long n) {
99         vector<ll> a(100), b(100);
100         a[0] = 2, a[1] = 6;
101         b[0] = 7, b[1] = 35;
102         for (int i = 2; i < 100; ++i) {
103             a[i] = (a[i - 1] * 2 + 3 * a[i - 2]) % mod;
104             b[i] = (b[i - 1] * 3 + 10 * b[i - 2]) % mod;
105         }
106         vector<ll> c(100);
107         for (int i = 0; i < 100; ++i) c[i] = a[i] * b[i] % mod;

```

```

108         return linear_seq::gao(c, n - 1);
109     }
110 };

```

```

1  ll qpow(ll a, ll b, ll p = mod) {
2      ll res = 1;
3      a %= p;
4      while (b > 0) {
5          if (b & 1) res = res * a % p;
6          a = a * a % p;
7          b >>= 1;
8      }
9      return res;
10 }
11
12 struct ChothollyTree {
13     map<int, ll> odt;
14
15     auto split(int x) {
16         auto it = prev(odt.upper_bound(x));
17         if (it->first != x) {
18             odt[x] = it->second;
19             ++it;
20         }
21         return it;
22     }
23
24     void add(int l, int r, ll v) {
25         auto end = split(r + 1);
26         for (auto it = split(l); it != end; it++)
27             it->second += v;
28     }
29
30     void assign(int l, int r, ll v) {
31         auto end = split(r + 1), begin = split(l);
32         odt.erase(begin, end);
33         odt[l] = v;
34     }
35
36     ll kth(int l, int r, int k) {
37         auto itr = split(r + 1), itl = split(l);
38         vector<pair<ll, ll> > V;
39         for (int R; itl != itr;) {
40             R = (itr--)->first;
41             V.emplace_back(itr->second, R - itr->first);
42         }

```

```

43         sort(V.begin(), V.end());
44         for (auto [x, y] : V) {
45             k -= y;
46             if (k <= 0) return x;
47         }
48     }
49
50     ll sum(int l, int r) {
51         ll res = 0;
52         auto itr = split(r + 1), itl = split(l);
53         for (int R; itl != itr;) {
54             R = (itr--)->first;
55             res = (res + itr->second * (R - itr->first));
56         }
57         return res;
58     }
59
60     ll power(int l, int r, ll x, ll y) {
61         ll res = 0;
62         auto itr = split(r + 1), itl = split(l);
63         for (int R; itl != itr;) {
64             R = (itr--)->first;
65             res = (res + qpow(itr->second, x, y) * (R - itr->first)) % y;
66         }
67         return res;
68     }
69 };

```

```

1  struct node {
2      int l, r;
3      mutable ll v;
4      node(const int &l, const int &r, const ll&v) : l(l), r(r), v(v) {}
5      inline bool operator < (const node &T) const {
6          return l < T.l;
7      }
8  };
9
10 set<node> odt;
11
12 auto split(int x) {
13     if (x > n) return odt.end();
14     auto it = prev(odt.upper_bound({x, 0, 0}));
15     if (it->l == x) return it;
16     int l = it->l, r = it->r;
17     ll v = it->v;
18     odt.erase(it);
19     odt.insert({l, x - 1, v});
20     return odt.insert({x, r, v}).first;

```

```

21 }
22
23 void assign(int l, int r, ll v) {
24     auto itr = split(r + 1), itl = split(l);
25     odt.erase(itl, itr);
26     odt.insert({l, r, v});
27 }
28
29 void add(int l, int r, ll v) {
30     auto end = split(r + 1);
31     for (auto it = split(l); it != end; it++) {
32         it->v += v;
33     }
34 }
35
36 ll kth(int l, int r, int k) {
37     auto itr = split(r + 1);
38     auto itl = split(l);
39     vector<pair<ll, ll>> V;
40     while (itl != itr) {
41         V.push_back({itl->v, itl->r - itl->l + 1});
42         itl++;
43     }
44     sort(V.begin(), V.end());
45     for (auto [x, y] : V) {
46         k -= y;
47         if (k <= 0) return x;
48     }
49 }
50
51 ll sum(int l, int r) {
52     ll ans = 0;
53     auto end = split(r + 1);
54     for (auto it = split(l); it != end; it++) {
55         ans += it->v;
56     }
57     return ans;
58 }
59
60 /*
61 i = l to r
62 ans = sum ai ^ x (mod y)
63 */
64 ll power(int l, int r, ll x, ll y) {
65     ll ans = 0;
66     auto end = split(r + 1);
67     for (auto it = split(l); it != end; it++) {
68         ans = (ans + qpow(it->v, x, y) * (it->r - it->l + 1)) % y;
69     }

```

```

70     return ans;
71 }

```

```

1  template<typename T>
2  T exgcd(T a, T b, T &x, T &y) {
3      if (!b) { x = 1, y = 0; return a; }
4      T d = exgcd(b, a % b, x, y);
5      T z = x;
6      x = y, y = z - y * (a / b);
7      return d;
8  }
9
10 template<typename T>
11 T inv(T a, T p) {
12     T x, y;
13     exgcd(a, p, x, y);
14     return (x % p + p) % p;
15 }
16
17 /*
18     arbitrarily i, j: gcd(m_i, m_j) = 1
19 */
20
21 ll CRT(const vector<ll> &r, const vector<ll> &m) {
22     assert(r.size() == m.size());
23     int n = int(r.size());
24     ll x = 0, p = 1;
25     for (int i = 0; i < n; i++) p *= m[i];
26     for (int i = 0; i < n; i++) {
27         ll t = p / m[i];
28         x = (x + r[i] * t * inv(t, m[i])) % p;
29     }
30     return x;
31 }

```

```

1  struct Deque {
2      int head, tail;
3      bool ok;
4      vector<int> q;
5
6      /*
7          init(size)
8          space : O(2 * n)
9      */
10
11     Deque(int n): head(n), tail(n - 1), ok(false), q(2 * n + 1, 0) {}
12
13     bool empty() {

```

```

14         return tail < head;
15     }
16
17     int size() {
18         return tail - head + 1;
19     }
20
21     int front() {
22         return ok ? q[tail] : q[head];
23     }
24
25     int back() {
26         return ok ? q[head] : q[tail];
27     }
28
29     void push_front(int x) {
30         ok ? q[++tail] = x : q[--head] = x;
31     }
32
33     void push_back(int x) {
34         ok ? q[--head] = x : q[++tail] = x;
35     }
36
37     void pop_front() {
38         ok ? tail-- : head++;
39     }
40
41     void reverse() {
42         ok ^= 1;
43     }
44 };

```

```

1  /*
2      O(N^2)
3  */
4  int n, m, e[N][N];
5  ll d[N];
6  bool vis[N];
7
8  void dijkstra(int s) {
9      memset(vis, 0, sizeof(vis));
10     memset(d, 0x3f, sizeof(d));
11     d[s] = 0;
12     for (int i = 1; i <= n; i++) {
13         int u = -1;
14         for (int v = 1; v <= n; v++) if (!vis[v] && (u == -1 || d[v] <
d[u])) u = v;

```

```

15         for (int v = 1; v <= n; v++) if (d[v] > d[u] + e[u][v]) d[v] = d[u]
+ e[u][v];
16     }
17 }

```

```

1 struct dijkstra {
2     int n;
3     vector<vector<pair<int, int>>> G;
4     vector<bool> vis;
5     vector<ll> d;
6
7     dijkstra(int n, vector<vector<pair<int, int>>> G) : n(n), d(n + 1),
vis(n + 1), G(G) {}
8
9     void dij(int s) {
10         d.assign(n + 1, INF);
11         vis.assign(n + 1, false);
12         priority_queue<pair<ll, int>, vector<pair<ll, int>>,
greater<pair<ll, int>>> q;
13         q.push({d[s] = 0, s});
14         while (!q.empty()) {
15             auto[_ , u] = q.top();
16             q.pop();
17             if (vis[u]) continue;
18             vis[u] = true;
19             for (auto[v, w] : G[u]) {
20                 if (d[v] > d[u] + w) {
21                     q.push({d[v] = d[u] + w, v});
22                 }
23             }
24         }
25     }
26 };

```

```

1 struct Dijkstra {
2     struct edge {
3         int node = -1;
4         int64_t weight = 0;
5
6         edge() {}
7
8         edge(int _node, int64_t _weight) : node(_node), weight(_weight) {}
9     };
10
11     struct state {
12         int64_t dist;
13         int node;
14     };

```

```

15         state() {}
16
17         state(int64_t _dist, int _node) : dist(_dist), node(_node) {}
18
19         bool operator<(const state &other) const {
20             return dist > other.dist;
21         }
22     };
23
24     int n;
25     vector<vector<edge>> adj;
26     vector<int64_t> dist;
27     vector<int> parent;
28
29     Dijkstra(int _n = 0) {
30         init(_n);
31     }
32
33     void init(int _n) {
34         n = _n;
35         adj.assign(n, {});
36     }
37
38     void add_directional_edge(int a, int b, int64_t weight) {
39         adj[a].emplace_back(b, weight);
40     }
41
42     void add_bidirectional_edge(int a, int b, int64_t weight) {
43         add_directional_edge(a, b, weight);
44         add_directional_edge(b, a, weight);
45     }
46
47     void dijkstra_check(priority_queue<state> &pq, int node, int from,
int64_t new_dist) {
48         if (new_dist < dist[node]) {
49             dist[node] = new_dist;
50             parent[node] = from;
51             pq.emplace(new_dist, node);
52         }
53     }
54
55     void dijkstra(const vector<int> &source) {
56         if (n == 0) return;
57         dist.assign(n, INF64);
58         parent.assign(n, -1);
59         priority_queue<state> pq;
60
61         for (int src : source)
62             dijkstra_check(pq, src, -1, 0);

```



```

63
64     while (!pq.empty()) {
65         state top = pq.top();
66         pq.pop();
67
68         if (top.dist > dist[top.node])
69             continue;
70
71         for (edge &e : adj[top.node])
72             dijkstra_check(pq, e.node, top.node, top.dist + e.weight);
73     }
74 }
75 };

```

```

1  #include<bits/stdc++.h>
2
3  #define se second
4  #define fi first
5  using namespace std;
6  typedef long long ll;
7  typedef pair<int, int> P;
8  const double eps = 1e-8;
9  const int NINF = 0xc0c0c0c0;
10 const ll INF = 0x3f3f3f3f3f3f3f3f;
11 const ll mod = 1e9 + 7;
12 const ll maxn = 1e6 + 5;
13 const int N = 1e5 + 5;
14
15 int n, m, s;
16
17 struct edge {
18     int to, cost;
19 };
20 vector<edge> G[N];
21 ll d[N];
22 int path[N];
23
24 void dijkstra(const int &s, const int &V) {
25     priority_queue<P, vector<P>, greater<P> > q;
26     memset(d, 0x3f, sizeof(d));
27     memset(path, -1, sizeof(path));
28     d[s] = 0;
29     q.push({0, s});
30     while (!q.empty()) {
31         P t = q.top();
32         q.pop();
33         int v = t.se;
34         if (d[v] < t.fi) continue;

```

```

35         for (int i = 0; i < G[v].size(); i++) {
36             edge e = G[v][i];
37             if (d[e.to] > d[v] + e.cost) {
38                 d[e.to] = d[v] + e.cost;
39                 path[e.to] = v;
40                 q.push({d[e.to], e.to});
41             }
42         }
43     }
44 }
45
46 void print(int e) {
47     if (d[e] >= INF) {
48         cout << -1 << '\n';
49         return;
50     }
51     vector<int> prev;
52     for (; e != -1; e = path[e]) prev.push_back(e);
53     reverse(prev.begin(), prev.end());
54     for (auto c:prev) cout << c << " ";
55     cout << '\n';
56 }
57
58 int main() {
59     ios::sync_with_stdio(false);
60     cin.tie(0);
61     cin >> n >> m;
62     for (int i = 1; i <= m; i++) {
63         int u, v, w;
64         cin >> u >> v >> w;
65         G[u].push_back({v, w});
66         G[v].push_back({u, w});
67     }
68     dijkstra(1, n);
69     print(n);
70     return 0;
71 }

```

```

1  /*
2      N = 点数 + 1
3  */
4
5  template<typename T>
6  struct Dinic {
7      struct Edge {
8          int from, to;
9          T cap, flow;
10     };

```

```

11
12     int n, m, s, t;
13     vector<int> d, cur;
14     vector <vector<int>> G;
15     vector<bool> vis;
16     vector <Edge> edges;
17
18     Dinic() : n(0) {}
19
20     Dinic(int n) : n(n), d(n, 0), cur(n, 0), G(n), vis(n, false) {}
21
22     void init(int n) {
23         for (int i = 0; i < n; i++) G[i].clear();
24         edges.clear();
25     }
26
27     void add(int from, int to, T cap, T incap = 0) {
28         edges.push_back({from, to, cap, 0});
29         edges.push_back({to, from, incap, 0});
30         m = edges.size();
31         G[from].push_back(m - 2);
32         G[to].push_back(m - 1);
33     }
34
35     bool bfs() {
36         vis.assign(n, false);
37         queue<int> Q;
38         d[s] = 0;
39         vis[s] = true;
40         Q.push(s);
41         while (!Q.empty()) {
42             int u = Q.front();
43             Q.pop();
44             for (auto v : G[u]) {
45                 Edge &e = edges[v];
46                 if (!vis[e.to] && e.cap > e.flow) {
47                     vis[e.to] = true;
48                     d[e.to] = d[u] + 1;
49                     Q.push(e.to);
50                 }
51             }
52         }
53         return vis[t];
54     }
55
56     T dfs(int u, T w) {
57         if (u == t || w == 0) return w;
58         T flow = 0, f;
59         for (int &i = cur[u]; i < G[u].size(); i++) {

```

```

60         int v = G[u][i];
61         Edge &e = edges[v];
62         if (d[e.to] == d[u] + 1 && (f = dfs(e.to, min(w, e.cap -
e.flow))) > 0) {
63             e.flow += f;
64             edges[v ^ 1].flow -= f;
65             flow += f;
66             w -= f;
67             if (w == 0) break;
68         }
69     }
70     return flow;
71 }
72
73 T Maxflow(int s, int t) {
74     return Maxflow(s, t, numeric_limits<T>::max());
75 }
76
77 T Maxflow(int s, int t, T limit) {
78     this->s = s;
79     this->t = t;
80     T maxflow = 0;
81     while (bfs()) {
82         cur.assign(n, 0);
83         maxflow += dfs(s, limit);
84     }
85     return maxflow;
86 }
87 };

```

```

1  #define maxn 250
2  #define INF 0x3f3f3f3f
3
4  struct Edge {
5      int from, to, cap, flow;
6      Edge(int u, int v, int c, int f) : from(u), to(v), cap(c), flow(f) {}
7  };
8
9  struct Dinic {
10     int n, m, s, t;
11     vector<Edge> edges;
12     vector<int> G[maxn];
13     int d[maxn], cur[maxn];
14     bool vis[maxn];
15
16     void init(int n) {
17         for (int i = 0; i < n; i++) G[i].clear();
18         edges.clear();

```

```

19     }
20
21     void AddEdge(int from, int to, int cap) {
22         edges.push_back(Edge(from, to, cap, 0));
23         edges.push_back(Edge(to, from, 0, 0));
24         m = edges.size();
25         G[from].push_back(m - 2);
26         G[to].push_back(m - 1);
27     }
28
29     bool BFS() {
30         memset(vis, 0, sizeof(vis));
31         queue<int> Q;
32         Q.push(s);
33         d[s] = 0;
34         vis[s] = 1;
35         while (!Q.empty()) {
36             int x = Q.front();
37             Q.pop();
38             for (int i = 0; i < G[x].size(); i++) {
39                 Edge& e = edges[G[x][i]];
40                 if (!vis[e.to] && e.cap > e.flow) {
41                     vis[e.to] = 1;
42                     d[e.to] = d[x] + 1;
43                     Q.push(e.to);
44                 }
45             }
46         }
47         return vis[t];
48     }
49
50     int DFS(int x, int a) {
51         if (x == t || a == 0) return a;
52         int flow = 0, f;
53         for (int& i = cur[x]; i < G[x].size(); i++) {
54             Edge& e = edges[G[x][i]];
55             if (d[x] + 1 == d[e.to] && (f = DFS(e.to, min(a, e.cap - e.flow))) >
0) {
56                 e.flow += f;
57                 edges[G[x][i] ^ 1].flow -= f;
58                 flow += f;
59                 a -= f;
60                 if (a == 0) break;
61             }
62         }
63         return flow;
64     }
65
66     int Maxflow(int s, int t) {

```

```

67     this->s = s;
68     this->t = t;
69     int flow = 0;
70     while (BFS()) {
71         memset(cur, 0, sizeof(cur));
72         flow += DFS(s, INF);
73     }
74     return flow;
75 }
76 };

```

```

1  /*
2      Hash S(s);
3      Sometimes we need to use a lot of primes to hash instead of overflow.
4
5      if MLE, try:
6          using ull = unsigned int;
7  */
8
9  struct Hash_string{
10     using ull = unsigned long long;
11     vector<ull> H, P;
12     ull base = 131;
13     int n;
14
15     Hash_string(string s):n((int)s.size()), H((int)s.size() + 1, 0),
P((int)s.size() + 1, 0){
16         P[0] = 1;
17         s = " " + s;
18         for (int i = 1; i <= n; i++) {
19             H[i] = H[i - 1] * base + s[i] - 'a' + 1;
20             P[i] = P[i - 1] * base;
21         }
22     }
23
24     ull get(int L, int R) {
25         return H[R] - H[L - 1] * P[R - L + 1];
26     }
27 };
28
29
30
31 /*
32     the only difference with last is get return pair to check.
33
34     Hash S(s);
35     Sometimes we need to use a lot of primes to hash instead of overflow.
36

```



```

15         }
16         to[++tot] = x;
17         cnt[tot] = 1;
18         next[tot] = head[k];
19         head[k] = tot;
20     }
21
22     inline int query(int x) {
23         int k = x % mod;
24         for (int i = head[k]; i; i = next[i]) if (to[i] == x) return
cnt[i];
25         return 0;
26     }
27 }
28
29 template<typename T> struct Hash {
30     int n, p, N;
31     vector<vector<T>> G;
32     vector<T> cnt, value;
33
34     Hash(int N, int p) : n(0), p(p), G(p), cnt(N), value(N) {}
35
36     void clear() {
37         n = 0;
38         vector<vector<T>> X;
39         swap(G, X);
40         vector<T> x1, x2;
41         swap(cnt, x1);
42         swap(value, x2);
43     }
44
45     void insert(int x) {
46         int k = x % p;
47         for (auto i : G[k]) {
48             if (value[i] == x) {
49                 cnt[i]++;
50                 return;
51             }
52         }
53         value[n] = x;
54         cnt[n] = 1;
55         G[k].push_back(n++);
56     }
57
58     int query(int x) {
59         int k = x % p;
60         for (auto i : G[k]) {
61             if (value[i] == x) {
62                 return cnt[i];

```



```

63         }
64     }
65     return 0;
66 }
67 };

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 1e6 + 5;
15
16 string s, t;
17 int n, m, kmp[N], nx[N];
18
19 void get_nextval() {
20     int i = 0, j = -1, len2 = t.size();
21     kmp[0] = -1;
22     while (i < len2) {
23         if (j == -1 || t[i] == t[j]) {
24             ++i, ++j;
25             if (t[i] != t[j]) kmp[i] = j;
26             else kmp[i] = kmp[j];
27         } else j = kmp[j];
28     }
29 }
30
31 void get_next() {
32     int i = 0, j = -1, len2 = t.size();
33     nx[0] = -1;
34     while (i < len2) {
35         if (j == -1 || t[i] == t[j]) nx[++i] = ++j;
36         else j = nx[j];
37     }
38 }
39
40 void get_kmp() {
41     int i = 0, j = 0, len1 = s.size(), len2 = t.size();
42     while (i < len1) {

```

```

43         if (j == -1 || s[i] == t[j]) i++, j++;
44         else j = kmp[j];
45         if (j == len2) cout << i - j + 1 << '\n', j = kmp[j];
46     }
47 }
48
49 int main() {
50     ios::sync_with_stdio(false);
51     cin.tie(nullptr);
52     cin >> s >> t;
53     n = s.size(), m = t.size();
54     get_nextval();
55     get_kmp();
56     get_next();
57     for (int i = 1; i <= m; i++) cout << nx[i] << " \n"[i == m];
58     return 0;
59 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 5e5 + 5;
15
16 int n, m, s, deep[N], f[N][21], lg[N];
17 vector<int> G[N];
18
19 void dfs(int u, int fa) {
20     f[u][0] = fa;
21     deep[u] = deep[fa] + 1;
22     for (auto v:G[u]) {
23         if (v == fa) continue;
24         dfs(v, u);
25     }
26 }
27
28 void ST(int n) {
29     for (int j = 1; (1 << j) <= n; j++)
30         for (int i = 1; i <= n; i++)

```

```

31         f[i][j] = f[f[i][j - 1]][j - 1];
32     }
33
34     int LCA(int u, int v) {
35         if (deep[u] < deep[v]) swap(u, v);
36         int h = deep[u] - deep[v];
37         for (int i = 0; i < 20; i++)
38             if (h & (1 << i)) u = f[u][i];
39         if (u == v) return u;
40         for (int i = 19; i >= 0; i--) {
41             if (f[u][i] != f[v][i]) {
42                 u = f[u][i];
43                 v = f[v][i];
44             }
45         }
46         return f[u][0];
47     }
48
49     void init() {
50         for (int i = 1; i <= n; i++)
51             lg[i] = lg[i - 1] + (1 << lg[i - 1] == i);
52     }
53
54     int main() {
55         ios::sync_with_stdio(false);
56         cin.tie(nullptr);
57         cin >> n >> m >> s;
58         for (int i = 1; i < n; i++) {
59             int u, v;
60             cin >> u >> v;
61             G[u].emplace_back(v);
62             G[v].emplace_back(u);
63         }
64         dfs(s, 0);
65         ST(n);
66         while (m--) {
67             int u, v;
68             cin >> u >> v;
69             cout << LCA(u, v) << '\n';
70         }
71         return 0;
72     }

```

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  const int INF = 0x3f3f3f3f;

```

```

6  const ll mod = 1e9 + 7;
7  const ll MAXN = 1e6 + 5;
8
9  ll p = mod;
10
11 inline ll qpow(ll a, ll b) {
12     ll base = a % p;
13     ll ans = 1;
14     while (b > 0) {
15         if (b & 1) ans = (ans * base) % p;
16         base = base * base % p;
17         b >>= 1;
18     }
19     return ans;
20 }
21
22 inline ll C(ll n, ll m) {
23     if (n < m) return 0; //组合数n<m特判
24     if (m > n - m) m = n - m; //组合数性质
25     ll a = 1, b = 1;
26     for (int i = 0; i < m; i++) {
27         a = (a * (n - i) % p); //组合数分子 a
28         b = (b * (i + 1)) % p; //组合数分母 b
29     }
30     return a * qpow(b, p - 2) % p; //费马小定理 a/b=a*b^(p-2)
31 }
32
33 inline ll Lucas(ll n, ll m, ll p = mod) {
34     return m == 0 ? 1 : Lucas(n / p, m / p) * C(n % p, m % p) % p;
35 }
36
37 int main() {
38     ios::sync_with_stdio(false);
39     cin.tie(0);
40     ll n, a, b;
41     cin >> n >> a >> b;
42     cout << (mod + qpow(2, n) - 1 - (C(n, a) + C(n, b)) % mod) % mod;
43     return 0;
44 }

```

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> P;
6  const double eps = 1e-8;
7  const int NINF = 0xc0c0c0c0;
8  const int INF = 0x3f3f3f3f;

```

```

9  const ll mod = 1e9 + 7;
10 const ll maxn = 3e7 + 5;
11
12 char s[maxn], str[maxn];
13 int n, p[maxn], ans;
14
15 void manacher_init() {
16     str[0] = str[1] = '#';
17     for (int i = 0; i < n; i++) {
18         str[(i << 1) + 2] = s[i];
19         str[(i << 1) + 3] = '#';
20     }
21     n = (n << 1) + 2;
22     str[n] = 0;
23 }
24
25 void manacher() {
26     int mx = 0, mid;
27     for (int i = 1; i < n; i++) {
28         if (mx > i) p[i] = min(p[(mid << 1) - i], p[mid] + mid - i);
29         else p[i] = 1;
30         for (; str[i + p[i]] == str[i - p[i]]; p[i]++);
31         if (p[i] + i > mx) mx = p[i] + i, mid = i;
32     }
33 }
34
35 int main() {
36     ios::sync_with_stdio(false);
37     cin.tie(0);
38     cin >> s;
39     n = strlen(s);
40     manacher_init();
41     manacher();
42     cout << *max_element(p, p + n) - 1 << '\n';
43     return 0;
44 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11

```

```

12 using namespace std;
13 using ll = long long;
14 using ull = unsigned long long;
15 using pii = pair<int, int>;
16
17 constexpr double eps = 1e-8;
18 constexpr int NINF = 0xc0c0c0c0;
19 constexpr int INF = 0x3f3f3f3f;
20 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
21 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
22 constexpr ll mod = 1e9 + 7;
23 constexpr ll N = 1e6 + 5;
24
25 int n, a[] = {0, 9, 82, 5, 3, 9, 2, 1, 0}, b[N];
26
27 void merge_sort(int L, int R){
28     if(L == R) return;
29     int mid = (L + R) >> 1;
30     merge_sort(L, mid);
31     merge_sort(mid + 1, R);
32     int i = L, j = mid + 1, k = L;
33     while(i <= mid && j <= R){
34         if(a[i] <= a[j]) b[k++] = a[i++];
35         else b[k++] = a[j++];
36     }
37     while(i <= mid) b[k++] = a[i++];
38     while(j <= R) b[k++] = a[j++];
39     for(int i = L; i <= R; i++) a[i] = b[i];
40 }
41
42 int main(){
43     ios::sync_with_stdio(0);
44     cin.tie(0);
45     int n = sizeof(a) / sizeof(a[0]);
46     cout << "n = " << n << '\n';
47     merge_sort(0, n - 1);
48     for(int i = 0; i < n; i++) cout << a[i] << " \n"[i == n-1];
49     return 0;
50 }

```

```

1 ll qpow(ll a, ll b, ll p) {
2     ll ans = 1;
3     while (b > 0) {
4         if (b & 1) ans = (__int128) ans * a % p;
5         a = (__int128) a * a % p;
6         b >>= 1;
7     }
8     return ans;

```

```

9   }
10
11  bool isPrime(ll x) {
12      if (x < 3) return x == 2;
13      if (x % 2 == 0) return false;
14      ll a[] = {2, 325, 9375, 28178, 450775, 9780504, 1795265022}, d = x - 1,
r = 0;
15      while (d % 2 == 0) d /= 2, ++r;
16      for (int i = 0; i < 7; i++) {
17          ll v = qpow(a, d, x);
18          if (v <= 1 || v == x - 1) continue;
19          for (int i = 0; i < r; i++) {
20              v = (__int128) v * v % x;
21              if (v == x - 1) {
22                  v = 1;
23                  break;
24              }
25              if (v == 1) return false;
26          }
27          if (v != 1) return false;
28      }
29      return true;
30  }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define lowbit(x) ((x) & -(x))
7
8  using namespace std;
9  using ll = long long;
10 using pii = pair<int, int>;
11
12 constexpr double eps = 1e-8;
13 constexpr int NINF = 0xc0c0c0c0;
14 constexpr int INF = 0x3f3f3f3f;
15 constexpr ll mod = 1e9 + 7;
16 constexpr ll N = 2e5 + 5;
17
18 #define ls (p << 1)
19 #define rs (ls | 1)
20 #define tm ((tl + tr) >> 1)
21
22 int tag[N << 2], sum[N << 2], a[N];
23 string s, f;
24

```

```

25 void build(int p, int tl, int tr) {
26     tag[p] = -1;
27     if (tl == tr) sum[p] = a[tl];
28     else {
29         build(ls, tl, tm);
30         build(rs, tm + 1, tr);
31         sum[p] = sum[ls] + sum[rs];
32     }
33 }
34
35 void modify(int p, int tl, int tr, int v) {
36     sum[p] = (tr - tl + 1) * v;
37     tag[p] = v;
38 }
39
40 void push_down(int p, int tl, int tr) {
41     if (tag[p] != -1) {
42         modify(ls, tl, tm, tag[p]);
43         modify(rs, tm + 1, tr, tag[p]);
44         tag[p] = -1;
45     }
46 }
47
48 int rangesum(int p, int tl, int tr, int L, int R) {
49     if (tl >= L && tr <= R) return sum[p];
50     push_down(p, tl, tr);
51     int res = 0;
52     if (L <= tm) res += rangesum(ls, tl, tm, L, R);
53     if (R > tm) res += rangesum(rs, tm + 1, tr, L, R);
54     return res;
55 }
56
57 void modify(int p, int tl, int tr, int L, int R, int v) {
58     if (tl >= L && tr <= R) return modify(p, tl, tr, v);
59     push_down(p, tl, tr);
60     if (L <= tm) modify(ls, tl, tm, L, R, v);
61     if (R > tm) modify(rs, tm + 1, tr, L, R, v);
62     sum[p] = sum[ls] + sum[rs];
63 }
64
65 bool check() {
66     int n, q;
67     cin >> n >> q;
68     cin >> s >> f;
69     s = " " + s, f = " " + f;
70     for (int i = 1; i <= n; i++) a[i] = f[i] - '0';
71     build(1, 1, n);
72     vector<int> l(q), r(q);
73     for (int i = 0; i < q; i++) {

```



```

74         cin >> l[i] >> r[i];
75     }
76     for (int i = q - 1; ~i; i--) {
77         int s = rangesum(1, 1, n, l[i], r[i]);
78         if (s == r[i] - l[i] + 1 - s) {
79             return false;
80         }
81         modify(1, 1, n, l[i], r[i], s > r[i] - l[i] + 1 - s);
82         s = rangesum(1, 1, n, l[i], r[i]);
83     }
84     for (int i = 1; i <= n; i++) {
85         if (rangesum(1, 1, n, i, i) != s[i] - '0') {
86             return false;
87         }
88     }
89     return true;
90 }
91
92 int main() {
93     ios::sync_with_stdio(false);
94     cin.tie(nullptr);
95
96     int T;
97     cin >> T;
98     while (T--) {
99         cout << (check() ? "YES" : "NO") << '\n';
100     }
101
102     return 0;
103 }

```

```

1  #define ls (p << 1)
2  #define rs (ls | 1)
3  #define tm ((tl + tr) >> 1)
4
5  int mx[N << 2], a[N << 2];
6
7  void up(int p) {
8      mx[p] = max(mx[ls], mx[rs]);
9  }
10
11 void build(int p, int tl, int tr) {
12     if (tl == tr) {
13         mx[p] = a[tl];
14     } else {
15         build(ls, tl, tm);
16         build(rs, tm + 1, tr);
17         up(p);

```

```

18     }
19 }
20
21 void modify(int p, int tl, int tr, int pos, int v) {
22     if (tl == tr) {
23         mx[p] = v;
24     } else {
25         if (tm >= pos) modify(ls, tl, tm, pos, v);
26         else modify(rs, tm + 1, tr, pos, v);
27         up(p);
28     }
29 }
30
31 int query(int p, int tl, int tr, int l, int r) {
32     if (l <= tl && tr <= r) {
33         return mx[p];
34     } else {
35         int ans = INT_MIN;
36         if (tm >= l) ans = max(ans, query(ls, tl, tm, l, r));
37         if (tm < r) ans = max(ans, query(rs, tm + 1, tr, l, r));
38         return ans;
39     }
40 }

```

```

1  #define ls (p << 1)
2  #define rs (ls | 1)
3  #define tm ((tl + tr) >> 1)
4
5  int mi[N << 2], a[N << 2];
6
7  void up(int p) {
8      mi[p] = min(mx[ls], mx[rs]);
9  }
10
11 void build(int p, int tl, int tr) {
12     if (tl == tr) {
13         mi[p] = a[tl];
14     } else {
15         build(ls, tl, tm);
16         build(rs, tm + 1, tr);
17         up(p);
18     }
19 }
20
21 void modify(int p, int tl, int tr, int pos, int v) {
22     if (tl == tr) {
23         mi[p] = v;
24     } else {

```

```

25         if (tm >= pos) modify(ls, tl, tm, pos, v);
26         else modify(rs, tm + 1, tr, pos, v);
27         up(p);
28     }
29 }
30
31 int query(int p, int tl, int tr, int l, int r) {
32     if (l <= tl && tr <= r) {
33         return mi[p];
34     } else {
35         int ans = INT_MAX;
36         if (tm >= l) ans = min(ans, query(ls, tl, tm, l, r));
37         if (tm < r) ans = min(ans, query(rs, tm + 1, tr, l, r));
38         return ans;
39     }
40 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;
13 const ll mod = 1e9 + 7;
14 const ll N = 1e5 + 5;
15
16 ll n, m, a[N], f[N][25], Log2[N];
17
18 struct ST {
19     vector<int> Log2;
20     vector<vector<int>> f;
21     int n;
22
23     ST(int n): n(n), f(n + 1, vector<int> (25, 0)), Log2(n + 1) {}
24
25     void init(vector<int> a) {
26         for (int i = 2; i <= n; i++) Log2[i] = Log2[i / 2] + 1;
27         for (int i = 1; i <= n; i++) f[i][0] = a[i];
28         for (int j = 1; (1 << j) <= n; j++)
29             for (int i = 1; i + (1ll << j) - 1 <= n; i++)
30                 f[i][j] = max(f[i][j - 1], f[i + (1ll << (j - 1))][j - 1]);
31     }

```

```

32
33     inline ll query(int l, int r) {
34         if (l > r) return 0;
35         int k = Log2[r - l + 1];
36         return max(f[l][k], f[r - (1ll << k) + 1][k]);
37     }
38 };
39
40 void ST() {
41     for (int i = 2; i <= n; i++) Log2[i] = Log[i / 2] + 1;
42     for (int i = 1; i <= n; i++) f[i][0] = a[i];
43     for (int j = 1; (1 << j) <= n; j++)
44         for (int i = 1; i + (1ll << j) - 1 <= n; i++)
45             f[i][j] = max(f[i][j - 1], f[i + (1ll << (j - 1))][j - 1]);
46 }
47
48 inline ll query(int l, int r) {
49     if (l > r) return 0;
50     int k = Log2[r - l + 1];
51     return max(f[l][k], f[r - (1ll << k) + 1][k]);
52 }
53
54 int main() {
55     ios::sync_with_stdio(false);
56     cin.tie(nullptr);
57     cin >> n >> m;
58     for (int i = 1; i <= n; i++) cin >> a[i];
59     ST();
60     while (m--) {
61         int l, r;
62         cin >> l >> r;
63         cout << query(l, r) << '\n';
64     }
65     return 0;
66 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  typedef long long ll;
9  typedef pair<int, int> pii;
10 const double eps = 1e-8;
11 const int NINF = 0xc0c0c0c0;
12 const int INF = 0x3f3f3f3f;

```

```

13  const ll mod = 1e9 + 7;
14  const ll N = 1e6 + 5;
15
16  int low[N], dfn[N], bridge[N], cntBridge, dfsClock;
17  vector<int> G[N];
18
19  void tarjan(int u, int fa) {
20      f[u] = fa;
21      low[u] = dfn[u] = ++dfsClock;
22      for (auto v:G[u]) {
23          if (!dfn[v]) {
24              tarjan(v, u);
25              low[u] = min(low[u], low[v]);
26              if (low[v] > dfn[u]) bridge[v] = true, cntBridge++;
27          } else if (dfn[v] < dfn[u] && v != fa) low[u] = min(low[u],
dfn[v]);
28      }
29  }
30
31  int main() {
32      ios::sync_with_stdio(false);
33      cin.tie(nullptr);
34
35      return 0;
36  }

```

```

1  #include<bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  using namespace std;
7  typedef long long ll;
8  typedef pair<int, int> P;
9  const double eps = 1e-8;
10 const int NINF = 0xc0c0c0c0;
11 const int INF = 0x3f3f3f3f;
12 const ll mod = 1e9 + 7;
13 const ll N = 2e4 + 5;
14
15 int n, m, cnt, low[N], dfn[N], cut[N];
16 vector<int> G[N];
17
18 void tarjan(int x, int fa) {
19     low[x] = dfn[x] = ++cnt;
20     int son = 0;
21     for (auto c:G[x]) {
22         if (!dfn[c]) {

```

```

23         tarjan(c, fa);
24         low[x] = min(low[x], low[c]);
25         if (low[c] >= dfn[x] && x != fa) cut[x] = true;
26         if (x == fa) son++;
27     } else low[x] = min(low[x], dfn[c]);
28 }
29 if (son >= 2 && x == fa) cut[x] = true;
30 }
31
32 int main() {
33     ios::sync_with_stdio(false);
34     cin.tie(0);
35     cin >> n >> m;
36     for (int i = 1; i <= m; i++) {
37         int u, v;
38         cin >> u >> v;
39         G[u].push_back(v);
40         G[v].push_back(u);
41     }
42     for (int i = 1; i <= n; i++)
43         if (!dfn[i])
44             tarjan(i, i);
45     int tot = 0;
46     for (int i = 1; i <= n; i++)
47         if (cut[i])
48             tot++;
49     cout << tot << '\n';
50     for (int i = 1; i <= n; i++)
51         if (cut[i])
52             cout << i << " ";
53     return 0;
54 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6
7  using namespace std;
8  using ll = long long;
9  using pii = pair<int, int>;
10
11 constexpr double eps = 1e-8;
12 constexpr int NINF = 0xc0c0c0c0;
13 constexpr int INF = 0x3f3f3f3f;
14 constexpr ll mod = 1e9 + 7;
15 constexpr ll N = 2e5 + 5;

```

```

16
17 int n, m, M, zkw_mx[N << 2], zkw_cnt[N << 2];
18
19 inline void update(int x) {
20     int ls = x << 1;
21     int rs = ls | 1;
22     if (zkw_mx[ls] == zkw_mx[rs]) {
23         zkw_mx[x] = zkw_mx[ls];
24         zkw_cnt[x] = zkw_cnt[ls] + zkw_cnt[rs];
25     } else if (zkw_mx[ls] < zkw_mx[rs]) {
26         zkw_mx[x] = zkw_mx[rs];
27         zkw_cnt[x] = zkw_cnt[rs];
28     } else {
29         zkw_mx[x] = zkw_mx[ls];
30         zkw_cnt[x] = zkw_cnt[ls];
31     }
32 }
33
34 inline void update(int x, int val) {
35     zkw_mx[x += M] = val;
36     for (x >>= 1; x; x >>= 1) update(x);
37 }
38
39 inline void query(int l, int r) {
40     int x, mx = -1, cnt = 0;
41     for (l += M - 1, r += M + 1; l ^ r ^ 1; l >>= 1, r >>= 1) {
42         if (~l & 1) {
43             x = l ^ 1;
44             if (mx < zkw_mx[x]) {
45                 mx = zkw_mx[x];
46                 cnt = zkw_cnt[x];
47             } else if (mx == zkw_mx[x]) cnt += zkw_cnt[x];
48         }
49         if (r & 1) {
50             x = r ^ 1;
51             if (mx < zkw_mx[x]) {
52                 mx = zkw_mx[x];
53                 cnt = zkw_cnt[x];
54             } else if (mx == zkw_mx[x]) cnt += zkw_cnt[x];
55         }
56     }
57     cout << mx << ' ' << cnt << '\n';
58 }
59
60 int main() {
61     ios::sync_with_stdio(false);
62     cin.tie(nullptr);
63     cin >> n >> m;
64     for (M = 1; M <= n; M <= 1);

```

```

65     for (int i = 1; i <= n; i++) {
66         cin >> zkw_mx[M + i];
67         zkw_cnt[M + i] = 1;
68     }
69     for (int i = M - 1; i; i--) update(i);
70     while (m--) {
71         string op;
72         int x, y;
73         cin >> op >> x >> y;
74         if (op == "Ask") query(x, y);
75         else update(x, y);
76     }
77     return 0;
78 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define lowbit(x) ((x) & -(x))
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10
11 using namespace std;
12 using ll = long long;
13 using ull = unsigned long long;
14 using pii = pair<int, int>;
15
16 constexpr double eps = 1e-8;
17 constexpr int NINF = 0xc0c0c0c0;
18 constexpr int INF = 0x3f3f3f3f;
19 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
20 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
21 constexpr ll mod = 1e9 + 7;
22 constexpr ll N = 2e6 + 5;
23
24 struct line {
25     int l, r, h, op;
26
27     inline bool operator<(const line &T) const {
28         return h == T.h ? op > T.op : h < T.h;
29     }
30 } a[N];
31
32 int n, m, x[N], cnt[N << 2], len[N << 2];
33

```



```

34 void push_up(int p, int tl, int tr) {
35     if (cnt[p]) {
36         len[p] = x[tr] - x[tl];
37     } else {
38         len[p] = len[ls] + len[rs];
39     }
40 }
41
42 void modify(int p, int tl, int tr, int L, int R, int v) {
43     if (x[tr] <= L || R <= x[tl]) return;
44     if (L <= x[tl] && x[tr] <= R) {
45         cnt[p] += v;
46         push_up(p, tl, tr);
47         return;
48     }
49     if (L <= x[tm]) modify(ls, tl, tm, L, R, v);
50     if (R > x[tm]) modify(rs, tm, tr, L, R, v);
51     push_up(p, tl, tr);
52 }
53
54 int main() {
55     ios::sync_with_stdio(false);
56     cin.tie(nullptr);
57
58     cin >> n;
59     for (int i = 0; i < n; i++) {
60         int x1, y1, x2, y2;
61         cin >> x1 >> y1 >> x2 >> y2;
62         a[++m] = {x1, x2, y1, 1};
63         x[m] = x1;
64         a[++m] = {x1, x2, y2, -1};
65         x[m] = x2;
66     }
67     n = m;
68     sort(a + 1, a + 1 + n);
69     sort(x + 1, x + 1 + n);
70     m = unique(x + 1, x + 1 + n) - (x + 1);
71     ll ans = 0;
72     for (int i = 1; i < n; i++) {
73         modify(1, 1, m, a[i].l, a[i].r, a[i].op);
74         ans += 1ll * len[1] * (a[i + 1].h - a[i].h);
75     }
76     cout << ans << '\n';
77
78     return 0;
79 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using db = double;
14 using ll = long long;
15 using ull = unsigned long long;
16 using pii = pair<int, int>;
17
18 const db PI = acos(-1.0);
19 constexpr db EPS = 1e-8;
20 constexpr int NINF = 0xc0c0c0c0;
21 constexpr int INF = 0x3f3f3f3f;
22 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
23 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
24 constexpr ll MOD = 1e9 + 7;
25 constexpr ll N = 1e6 + 5;
26
27 int tot, rt[N];
28
29 struct node {
30     int l, r;
31     ll sum;
32 } t[N << 5];
33
34 void modify(int &x, int l, int r, int v) {
35     t[++tot] = t[x];
36     x = tot;
37     t[x].sum += v;
38     if (l == r) return;
39     int mid = (l + r) >> 1;
40     if (v <= mid) modify(t[x].l, l, mid, v);
41     else modify(t[x].r, mid + 1, r, v);
42 }
43
44 ll query(int x, int y, int l, int r, ll v) {
45     if (l == r) return t[y].sum - t[x].sum;
46     int mid = (l + r) >> 1;
47     if (v <= mid) return query(t[x].l, t[y].l, l, mid, v);
48     else return query(t[x].r, t[y].r, mid + 1, r, v) + t[t[y].l].sum -
t[t[x].l].sum;

```

```

49 }
50
51 int main() {
52     ios::sync_with_stdio(false);
53     cin.tie(nullptr);
54     cout << fixed << setprecision(20);
55
56     int n, m;
57     cin >> n >> m;
58     for (int i = 1; i <= n; i++) {
59         int x;
60         cin >> x;
61         modify(rt[i] = rt[i - 1], 1, 1000000000, x);
62     }
63     int l, r;
64     ll ans = 0;
65     while (m--) {
66         cin >> l >> r;
67         l = (l + ans) % n + 1;
68         r = (r + ans) % n + 1;
69         if (l > r) swap(l, r);
70         ans = 0;
71         while (true) {
72             ll res = query(rt[l - 1], rt[r], 1, 1000000000, ans + 1);
73             if (res == ans) break;
74             ans = res;
75         }
76         cout << ++ans << '\n';
77     }
78
79     return 0;
80 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using db = double;
14 using ll = long long;
15 using ull = unsigned long long;

```

```

16 using pii = pair<int, int>;
17
18 const db PI = acos(-1.0);
19 constexpr db EPS = 1e-8;
20 constexpr int NINF = 0xc0c0c0c0;
21 constexpr int INF = 1e9;
22 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
23 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
24 constexpr ll MOD = 1e9 + 7;
25 constexpr ll N = 1e6 + 5;
26
27 int rt[N], a[N], tot;
28
29 struct node {
30     int l, r, cnt;
31 } t[N << 5];
32
33 void modify(int &x, int l, int r, int v) {
34     t[++tot] = t[x];
35     x = tot;
36     t[x].cnt++;
37     if (l == r) return;
38     int mid = (l + r) >> 1;
39     if (v <= mid) modify(t[x].l, l, mid, v);
40     else modify(t[x].r, mid + 1, r, v);
41 }
42
43 int query(int x, int y, int l, int r, int v) {
44     if (l == r) return l;
45     int mid = (l + r) >> 1, cnt = t[t[y].l].cnt - t[t[x].l].cnt;
46     if (v <= cnt) return query(t[x].l, t[y].l, l, mid, v);
47     else return query(t[x].r, t[y].r, mid + 1, r, v - cnt);
48 }
49
50 int main() {
51     ios::sync_with_stdio(false);
52     cin.tie(nullptr);
53     cout << fixed << setprecision(20);
54
55     int n, m;
56     cin >> n >> m;
57     for (int i = 1; i <= n; i++) {
58         cin >> a[i];
59         modify(rt[i] = rt[i - 1], -INF, INF, a[i]);
60     }
61     while (m--) {
62         int l, r, k;
63         cin >> l >> r >> k;
64         cout << query(rt[l - 1], rt[r], -INF, INF, k) << '\n';

```

```

65     }
66
67     return 0;
68 }

```

```

1  #include <bits/stdc++.h>
2
3  #define fi first
4  #define se second
5  #define mp make_pair
6  #define pb push_back
7  #define ls (p << 1)
8  #define rs (ls | 1)
9  #define tm ((tl + tr) >> 1)
10 #define lowbit(x) ((x) & -(x))
11
12 using namespace std;
13 using db = double;
14 using ll = long long;
15 using ull = unsigned long long;
16 using pii = pair<int, int>;
17
18 const db PI = acos(-1.0);
19 constexpr db EPS = 1e-8;
20 constexpr int NINF = 0xc0c0c0c0;
21 constexpr int INF = 1000000;
22 constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
23 constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
24 constexpr ll MOD = 1e9 + 7;
25 constexpr ll N = 1e5 + 5;
26
27 struct node {
28     int l, r, cnt;
29     ll sum;
30 } t[N << 5];
31
32 int tot, rt[N];
33
34 void modify(int &x, int l, int r, int v) {
35     t[++tot] = t[x];
36     x = tot;
37     t[x].cnt++;
38     t[x].sum += v;
39     if (l == r) return;
40     int mid = (l + r) >> 1;
41     if (v <= mid) modify(t[x].l, l, mid, v);
42     else modify(t[x].r, mid + 1, r, v);
43 }

```

```

44
45 ll query(int x, int y, int l, int r, int v) {
46     if (l == r) return 1ll * v * l;
47     int mid = (l + r) >> 1, cnt = t[t[y].r].cnt - t[t[x].r].cnt;
48     if (v <= cnt) return query(t[x].r, t[y].r, mid + 1, r, v);
49     else return query(t[x].l, t[y].l, l, mid, v - cnt) + t[t[y].r].sum -
t[t[x].r].sum;
50 }
51
52 int main() {
53     ios::sync_with_stdio(false);
54     cin.tie(nullptr);
55     cout << fixed << setprecision(20);
56
57     int T;
58     cin >> T;
59     while (T--) {
60         tot = 0;
61         int n;
62         cin >> n;
63         for (int i = 1; i <= n; i++) {
64             int x;
65             cin >> x;
66             modify(rt[i] = rt[i - 1], 1, 1000000, x);
67         }
68         int m;
69         cin >> m;
70         while (m--) {
71             int l, r, k;
72             cin >> l >> r >> k;
73             int x = r - l + 1;
74             cout << 1ll * x * (x + 1) * (2 * x + 1) / 6 + query(rt[l - 1],
rt[r], 1, 1000000, k) << '\n';
75         }
76     }
77
78     return 0;
79 }

```