

Problem Menu

< Back

Statement

Submissions



Your score

not attempted

Spoilers

Show difficulty Show tags

solved by 2 / 5

Top users by score

#	User	Score
1	Zanite	100
2	Raphela	100
4	filiomerel	0
5	kaka_jayapura	0

Arkavidia 9.0 - Penyisihan CP > A

Submission #4625747

Arkavidia 9.0 - Penyisihan CP / A. Another Tree Cost Problem

Accepted • Raphela • C++17 • November 15, 2025 at 19:32:44

Test Data Results



```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int MAXN = 100000;
5 const int MAXK = 7;
6 const long long INF = (long long)4e18;
7
8 struct Edge {
9     int to;
10    long long w[MAXN];
11 };
12
13 struct Anc {
14     int cid;
15     long long dist[MAXN];
16 };
17
18 long long S[MAXN][MAXN + 5];
19 long long distR[MAXN][MAXN + 5];
20
21 bool removedNode[MAXN + 5];
22 int subSize[MAXN + 5];
23 vector<int> centroids;
24 vector<Anc> anc[MAXN + 5];
25
26 void dfsDistR(int v, int p) {
27     for (const auto &e : g[v]) {
28         int to = e.to;
29         if (to == p) continue;
30         for (int c = 0; c < K; ++c) {
31             distR[c][to] = distR[c][v] + e.w[c];
32         }
33         dfsDistR(to, v);
34     }
35 }
36
37 void dfsSize(int v, int p) {
38     subSize[v] = 1;
39     for (const auto &e : g[v]) {
40         int to = e.to;
41         if (to == p || removedNode[to]) continue;
42         dfsSize(to, v);
43         subSize[v] += subSize[to];
44     }
45
46     Anc a;
47     a.cid = cid;
48     for (int c = 0; c < K; ++c) a.dist[c] = distArr[c];
49     anc[v].push_back(a);
50     for (const auto &e : g[v]) {
51         int to = e.to;
52         if (to == p || removedNode[to]) continue;
53         long long nd[MAXN];
54         for (int c = 0; c < K; ++c) nd[c] = distArr[c] + e.w[c];
55         dfsDistCentroid(to, v, cid, nd);
56     }
57 }
58
59 void decompose(int v) {
60     dfsSize(v, 0);
61     int size = subSize[v];
62     int parent = 0;
63     int cur = v;
64     bool changed = true;
65     while (changed) {
66         changed = false;
67         for (const auto &e : g[cur]) {
68             int to = e.to;
69             if (to == parent || removedNode[to]) continue;
70             if (subSize[to] > size / 2) {
71                 parent = cur;
72                 cur = to;
73                 changed = true;
74             }
75         }
76         int cnode = cur;
77         int cid = (int)centroids.size();
78         centroids.push_back(cnode);
79         removedNode[cnode] = true;
80         long long distArr[MAXN];
81         for (int i = 0; i < K; ++i) distArr[i] = 0;
82         dfsDistCentroid(cnode, 0, cid, distArr);
83         for (const auto &e : g[cnode]) {
84             int to = e.to;
85             if (removedNode[to]) continue;
86             decompose(to);
87         }
88     }
89
90     long long F[MAXN][MAXN + 5];
91     long long Bval[MAXN + 5];
92     long long best[MAXN][MAXN + 5];
93
94     // ... (rest of the code) ...
95 }
```

```

101 int main() {
102     ios::sync_with_stdio(false);
103     cin.tie(nullptr);
104
105     if (!(cin >> N >> K >> R)) return 0;
106
107     for (int i = 0; i < N - 1; ++i) {
108         int U, V;
109         cin >> U >> V;
110         for (int c = 0; c < K; ++c) {
111             long long x;
112             cin >> x;
113             e1.w[c] = x;
114             e2.w[c] = x;
115         }
116         g[U].push_back(e1);
117         g[V].push_back(e2);
118     }
119
120     for (int c = 0; c < K; ++c) {
121         for (int j = 1; j <= N; ++j) {
122             long long x;
123             cin >> x;
124             S[c][j] = x;
125         }
126     }
127
128     dfsDistR(R, 0);
129
130     decompose(1);
131     int C = (int)centroids.size();
132
133     for (int c = 0; c < K; ++c) {
134         for (int cid = 0; cid < C; ++cid) {
135             best[c][cid] = INF;
136         }
137     }
138
139     long long val = distR[c][1];
140     long long q = INF;
141     for (const auto &a : anc[1]) {
142         int cid = a.cid;
143         long long cand = best[c][cid] + a.dist[c];
144         if (cand < q) q = cand;
145     }
146     if (q < val) val = q;
147     F[c][1] = val;
148
149     long long bi = INF;
150     for (int c = 0; c < K; ++c) {
151         long long cand = S[c][1] + F[c][1];
152         if (cand < bi) bi = cand;
153     }
154     Bval[1] = bi;
155     int cid = a.cid;
156     for (int c = 0; c < K; ++c) {
157         long long cand = Bval[1] + a.dist[c];
158         if (cand < best[c][cid]) best[c][cid] = cand;
159     }
160 }
161
162 for (int P = 1; P <= N; ++P) {
163     long long ans = INF;
164     for (int c = 0; c < K; ++c) {
165         if (F[c][P] < ans) ans = F[c][P];
166     }
167     if (P == R) ans = 0;
168     if (P > 1) cout << ' ';
169     cout << ans;
170 }
171
172 cout << '\n';
173
174 return 0;
175 }

```