

[AtCoder] ABC 087 D – People on a Line

2019年11月24日

スポンサーリンク

問題

D - People on a Line

AtCoder is a programming contest site for anyone from beginners to experts. We hold weekly programming contests.



 https://atcoder.jp/contests/abc087/tasks/arc090_b

方針

座標間の距離が与えられるので、相対的な位置が分かります。したがって、ある座標の値を $x_i = 0$ として、 M 個の情報に誤りがあるかどうかを調べます。

グラフ

各座標をグラフの頂点として、 M 個の辺からなるグラフを考えます。 M 個の情報から隣接リストを作成します。この隣接リストに座標間の距離の情報も追加しておきます。この座標間の距離は辺の重みではないことに注意します。例えば、 (L_i, R_i, D_i) という情報から、頂点 L_i の隣接する頂点は R_i であり、距離は D_i であり、頂点 R_i の隣接する頂点は L_i であり、距離は $-D_i$ であるという隣接リストが得られます。

幅優先探索

よくあるグラフの幅優先探索を行います。一度も訪れたことがない頂点 i の座標を $x_i = 0$ として、座標 i を含むグラフを走査します。このグラフを走査するなかで、座標の値を更新してき、誤りがあるかどうかを調べます。例えば、グラフ G の頂点が $(1, 2, 3)$ であり、点 1 に隣接する頂点が $2, 3)$ であり、頂点 2 に隣接する頂点が 3 であるとします。ここで、頂点 1 から幅優先探索を行うと、 x_2, x_3 の値が定まります。次に、頂点 2 から頂点 3 へ訪れたとき、 x_3 の値が更新されるとき誤りがあることになります。

コード

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4
5  struct Node {
6      int id, cost;
7      Node(int id, int cost) : id(id), cost(cost){}
8  };
9
10 int main() {
11     int N, M;
12     cin >> N >> M;
13     int L[M], R[M], D[M];
14     vector<Node> adj[N];
15     for (int i = 0; i < M; i++) {
16         cin >> L[i] >> R[i] >> D[i];
17         L[i]--;
18         R[i]--;
19         adj[L[i]].push_back(Node(R[i], D[i]));
20         adj[R[i]].push_back(Node(L[i], -D[i]));
21     }
22     int x[N];
23     int flag[N]{};
24     fill(x, x + N, INT32_MAX);
25     for (int i = 0; i < N; i++) {
26         if (flag[i] != 0) continue;
27         x[i] = 0;
28         deque<int> q;
29         q.push_back(i);
30         while (!q.empty()) {
31             int v = q.front();
32             flag[v] = 1;
33             q.pop_front();
34             for (int j = 0; j < adj[v].size(); j++) {
35                 int id = adj[v][j].id;
36                 int cost = adj[v][j].cost;
37                 if (x[id] == INT32_MAX) {
38                     x[id] = x[v] + cost;
39                     q.push_back(id);
40                     flag[id] = 1;
41                 } else {
42                     if (x[id] != x[v] + cost) {
43                         cout << "No\n";
44                         return 0;
45                     }
46                 }
47             }
48         }
49     }
50     cout << "Yes\n";
51     return 0;
52 }

```

AtCoder, グラフ理論, 幅優先探索, 探索

Posted by ヤマカサ