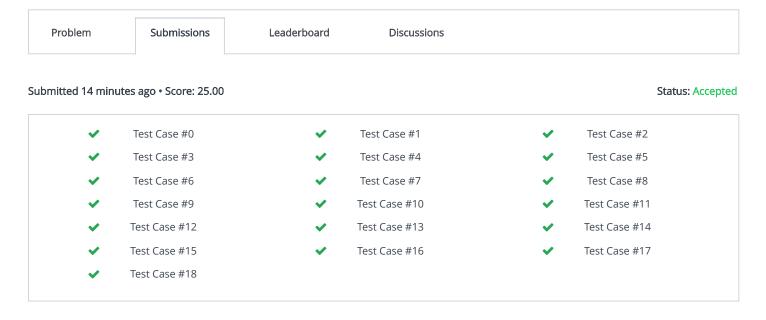
All Contests > Assignment 03 | Introduction to Algorithms | Batch 03 > Building

Building



Submitted Code

```
Language: C++20
                                                                                              P Open in editor
 1 #include <bits/stdc++.h>
 2 using namespace std;
 3
 4 #define pii pair<int, pair<int, int>>
 5 int n, m;
 6 vector<pii> ans;
 8 // DSU_UNION_BY_SIZE
 9 const int N = 1e5 + 5;
10 int parent[N];
11 int parentSize[N];
12
13 void dsu_set(int n)
14 {
       for (int i = 1; i <= n; i++)
15
16
           parent[i] = i;
17
           parentSize[i] = 1;
18
19
20 }
21
22 int dsu_find(int node)
23 {
24
       while (parent[node] != node)
25
       {
26
           node = parent[node];
27
```

```
28
       return node;
29 }
30
31 // This is union by size
32 void dsu_union(int a, int b)
33 {
34
       int leaderA = dsu_find(a);
       int leaderB = dsu_find(b);
35
       if (leaderA != leaderB)
36
37
           if (parentSize[leaderA] > parentSize[leaderB])
38
39
               // Leader is 'A'
40
               parent[leaderB] = leaderA;
41
               parentSize[leaderA] += parentSize[leaderB];
42
           }
43
44
45
           else
46
           {
47
               // Leader is 'B'
48
               parent[leaderA] = leaderB;
49
               parentSize[leaderB] += parentSize[leaderA];
50
           }
       }
51
52 }
53
54 int main()
55 {
56
       cin >> n >> m;
57
58
       priority_queue<pii, vector<pii>, greater<pii>> pq;
59
60
       dsu_set(n);
61
62
       for (int i = 1; i <= m; i++)
63
64
           int u, v, w;
           cin >> u >> v >> w;
65
66
           pq.push({w, {u, v}});
       }
67
68
       // Kruskal's Algorithm
69
70
       while (!pq.empty())
71
72
           auto el = pq.top();
73
           pq.pop();
74
75
           int w = el.first;
76
           int u = el.second.first;
77
           int v = el.second.second;
78
79
           int leaderU = dsu_find(u);
           int leaderV = dsu_find(v);
80
           if (leaderU == leaderV)
81
82
               continue;
           ans.push_back({w, {u, v}});
83
84
85
           dsu_union(u, v);
       }
86
87
88
       long long int cost = 0;
89
       for (auto e: ans)
90
       {
           // cout << e.second.first << " " << e.second.second << " " << e.first << endl;
91
           cost+= (long long int)(e.first);
92
       }
93
```

```
94
        if(ans.size()==n-1)
 95
         {
 96
             cout<<cost<<endl;</pre>
        }
 97
 98
        else
         cout<<"-1"<<endl;</pre>
 99
100
101
102
103
         return 0;
104 }
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |