1135. Connecting Cities With Minimum Cost

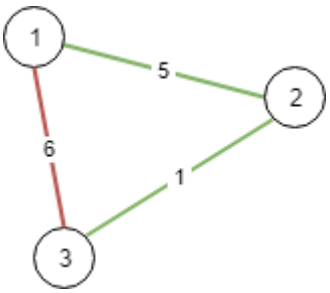
Medium 800 45 Add to List Share

There are `n` cities labeled from `1` to `n`. You are given the integer `n` and an array `connections` where `connections[i] = [x_i, y_i, cost_i]` indicates that the cost of connecting city `x_i` and city `y_i` (bidirectional connection) is `cost_i`.

Return *the minimum **cost** to connect all the `n` cities such that there is at least one path between each pair of cities*. If it is impossible to connect all the `n` cities, return `-1`,

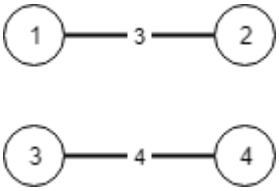
The **cost** is the sum of the connections' costs used.

Example 1:




```
Input: n = 3, connections = [[1,2,5],[1,3,6],[2,3,1]]
Output: 6
Explanation: Choosing any 2 edges will connect all cities so we
choose the minimum 2.
```

Example 2:




```
Input: n = 4, connections = [[1,2,3],[3,4,4]]
Output: -1
Explanation: There is no way to connect all cities even if all
edges are used.
```

```
1  public class Solution {
2
3      private int[] paren
4
5      private int Find(in
6      {
7          if(parent[x] ==
8              return x;
9
10         parent[x] =
   Find(parent[x]);
11         return parent[x
12     }
13
14     public int MinimumC
   n, int[][] connections)
15
16         //Sort by weigh
   of the edges
17         Array.Sort(conn
   Comparer<int[]>.Create(
18             return
   x[2].CompareTo(y[2]);
19         }));
20
21         parent = new in
22         for(int i = 1;
   i++)
23             parent[i] =
24
25         int edgesCount
26         int cost = 0;
27         foreach(int[] c
   in connections)
28         {
29             int u_rep =
   Find(connection[0]);
```

Testcase Run Code Result

Accepted Runtime: 129 ms

Your input: 3 [[1,2,5],[1,3,6]

Output: 6

Expected: 6

Console Use Example Testcase