Minimum Spanning Tree

Given a weighted, undirected and connected graph of **V** vertices and **E** edges. The task is to find the sum of weights of the edges of the Minimum Spanning Tree.

Example 1:

```
Input:
3 3
0 1 5
1 2 3
0 2 1

Output:
4
Explanation:


The Spanning Tree resulting in a weight
of 4 is shown above.
```

Example 2:

```
Input:
2 1
0 1 5

Output:
5
Explanation:
Only one Spanning Tree is possible
```

```
which has a weight of 5.
```

Your task:
Since this is a functional problem you don't have to worry about input, you just have to complete the function **spanningTree()** which takes number of vertices V and an adjacency matrix adj as input parameters and returns an integer denoting the sum of weights of the edges of the Minimum Spanning Tree. Here adj[i] contains a list of lists containing two integers where the first integer a[i][0] denotes that there is an edge between i and a[i][0][0] and second integer a[i][0][1] denotes that the distance between edge i and a[i][0][0] is a[i][0][1].

In other words , adj[i][j] is of form { u , wt } . So,this denotes that i th node is connected to u th node with edge weight equal to wt.

Expected Time Complexity: O(ElogV).
Expected Auxiliary Space: $O(V^2)$.

Constraints:
$2 \le V \le 1000$
$V-1 \le E \le (V*(V-1))/2$
$1 \le w \le 1000$
Graph is connected and doesn't contain self loops & multiple edges.