Instructions -> In code file, execute in dev by pressing F11.

Explaining code :

1. Get the information in input text to my variable, and arrays. (ne, e[ ], w[ ])
2. Searching all the rooms in e[ ], and continuously move the values into another array if the value is not existing in another array. If this action finished, we can get the array that contains all vertexes. We can also get the number of vertexes. (make v[ ]->nv)
3. Sorting the edges by sequence of weight. The edge which has smallest weight goes to first room of array. I make new array. (ew[ ])
4. I make forest array that first has the value(index+1), capacity is same as vertexes. I also make startedge, endedge.
5. Kruscal function : Checking all the edge from startedge to endedge. If we can connect edge, we make forest number of both vertexes of edge same. If not, just pass the edge.
6. If finished kruscal function but all forest number is not same, minimum spanning tree cannot be made. So print NULL. If not, print total weight.
7. Repeat the kruscal function to get second minimum spanning tree. But, when this time, exclude one edge of minimum spanning tree continuously.
8. Get the minimum weight of totals getting from 7.instruction. If not existing any total weight, print NULL.

Time complexity is O(n^2). Because the part which has the highest number of operation is 2-dimensional for-loop. ( Line 132 ~ 139 : number of edges \* number of edges )

Running time : 500 edges -> 0.218sec

777 edges -> 0.506sec