

Judgement Day 1

Problem Explanation:

We are given N temples and M roads connecting these temples. Your task is to find if it is possible to visit all temples before getting caught by the Azazel. If that is possible then you have to print the minimum time taken to visit the temples otherwise print -1.

Prerequisites

Minimum spanning tree. <https://www.ics.uci.edu/~eppstein/161/960206.html>

Approach

The question can be broken down into finding the minimum time taken to reach all the temples and then checking it with the time Azazel is going to reach to them i.e. T.

We have to now find the minimum time taken to reach all the temples which is same as to find the MST. The cost of MST will give us the minimum time it will take to reach all temples.

1. Create a graph using the given edge list. We will use prim's algorithm to find the minimum spanning tree.
2. Initialize the cost of all vertices as infinite.
3. Create an empty priority queue PQ. Every item of PQ is a pair (cost, vertex). Make sure to compare based on cost.
4. Initialize all vertices as not part of MST yet. We use a boolean array `visited` for this purpose. This array is required to make sure that an already considered vertex is not included in PQ again. This is where Prim's implementation differs from Dijkstra. In Dijkstra's algorithm, we didn't need this array as distances always increase. We require this array here because the cost of a processed vertex may decrease if not checked.
5. Insert source vertex into PQ and make its cost as 0.
6. Keep a variable `totalCost` = 0
7. While either PQ doesn't become empty
 - a. Extract minimum key vertex from PQ. Let the extracted vertex be u.
 - b. Include u in MST using visited[u] = true.
 - c. totalCost += cost[u]
 - d. Loop through all adjacent of u and do the following for every vertex v.
// If the cost of edge (u,v) is smaller than the cost of v and v is not already in MST

If visited[v] = false && cost[v] > weight(u, v)
 Update key of v, i.e., do cost[v] = weight(u, v)
 Insert v into the PQ

8. Check if `totalCost` < T:
 - a. If true, print `totalCost`
 - b. Else print -1