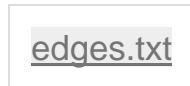


1
point

3. Question 3

In this programming problem you'll code up Prim's minimum spanning tree algorithm.

Download the text file below.



This file describes an undirected graph with integer edge costs. It has the format

```
[number_of_nodes] [number_of_edges]
```

```
[one_node_of_edge_1] [other_node_of_edge_1] [edge_1_cost]
```

```
[one_node_of_edge_2] [other_node_of_edge_2] [edge_2_cost]
```

```
...
```

For example, the third line of the file is "2 3 -8874", indicating that there is an edge connecting vertex #2 and vertex #3 that has cost -8874.

You should NOT assume that edge costs are positive, nor should you assume that they are distinct.

Your task is to run Prim's minimum spanning tree algorithm on this graph. You should report the overall cost of a minimum spanning tree --- an integer, which may or may not be negative --- in the box below.

IMPLEMENTATION NOTES: This graph is small enough that the straightforward $O(mn)$ time implementation of Prim's algorithm should work fine. OPTIONAL: For those of you seeking an additional challenge, try implementing a heap-based version. The simpler approach, which should already give you a healthy speed-up, is to maintain relevant edges in a heap (with keys = edge costs). The superior approach stores the unprocessed vertices in the heap, as described in lecture. Note this requires a heap that supports deletions, and you'll probably need to maintain some kind of mapping between vertices and their positions in the heap.