

Making Friends Report

Author 1 and Author 2

May 9, 2019

1 Results

Briefly comment the results, did the script say all your solutions were correct? Approximately how long time does it take for the program to run on the largest input? What takes the majority of the time?

All the tests are correct and take about 13 seconds to execute. The Huge data input executed in roughly 10 seconds. The majority of the time is spent on executing the large data input. This is to be expected since the huge data input is the largest input.

2 Implementation details

How did you implement the solution? Which data structures were used? Which modifications to these data structures were used? What is the overall running time? Why?

We used Prim's algorithm in order to create a MST. We have both a Node and an Edge class to represent them respectively, a class that parses and creates a graph from the input as well as a class that actually creates a MST from the parsed graph. We used a PriorityQueue to keep track of the least 'costly' unvisited edges which are put in a set to output the MST. we used a set to keep track of all the unvisited nodes. In the Graph class we use a map to hold all the nodes with their ID as key.

The time complexity of Prim's algorithm is $O(m \cdot \log(n))$. But our algorithm is $O((n-1) \cdot \log(m))$ in time complexity. You could argue that the time complexity should be $O(n \cdot (n-1) \cdot \log(n))$ since we have a time complexity of $O((n-1) \cdot \log(m))$ in our $O(n)$ loop. Since we can only add each edge in two ways we get $2 \cdot m \cdot \log(m)$ which simplify to the time complexity $O(m \cdot \log(n))$.

But $O(m \cdot \log(m))$ is really the same time complexity as $O(m \cdot \log(n))$ since, in worst case, $m = ((n-1) + (n-2) + \dots + (n-n)) = (n-1) \cdot n / 2 = n \cdot n$.