

Analysis:

Once you have completed and tested both implementations, compare the run time of each implementation on a file containing actor pair pairs that you generate yourself. Analyze the timing results for the two implementations and **write your analysis in the file `Report.pdf`**.

We weren't able to implement union.

In addition, **answer the following questions in `Report.pdf`**:

1. Which implementation is better and by how much?

BFS is better in this case because we are working with a static graph.

2. When does the union-find data structure significantly outperform BFS (if at all)?

If you have a graph where new edges are being added, then the union-find data structure would be faster to find any cycles in the graph or the minimum spanning tree.

3. What arguments can you provide to support your observations?

Union find is only helpful only when edges and vertices are not deleted.

Union find has time complexity of  $O(E \cdot \alpha(V) + V)$  and space complexity of  $O(V)$ . BFS has time complexity  $O(|V| + |E|)$  and space complexity of  $O(|V|)$ . Both the union find and BFS have similar time and space complexities, respectively. Since the time complexities are almost the same, the space requirement or implementation detail can make the difference between which method will run faster.