

1. Which implementation is better and by how much?

For 100 queries of the actor pair:

FRANCO, DAVE      KELLY, MATTHEW ROBERT

Breadth-First Search took **15771144321 ns** and the Union-Find implementation took **1163394 ns**. UF is considerably faster than BFS as BFS took over 13000x the time it took UF to run the same queries, this is of order of magnitude 4.

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

The Union-Find data structure is faster than the implementation of BFS when we can disregard the movies that connect any two actors, and furthermore the paths that connect the two actors in a search. This is suitable for querying the connectedness of two actors without considering how they are connected.

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

Searching if two actors are connected in a Union-Find data structure is nearly  $O(1)$ , represented as  $O(\log^*n)$ , after path compression. In our implementation we store the size of the set in the sentinel node to perform this optimization. To get the year that the two actors are join at, we must traverse up the tree until the first node that exists for both actors. As this common actor must be at the same height in the path for both queried actors, you only need to check once at each level. Therefore the runtime to get the year is similarly  $O(\log^*n)$ , or nearly  $O(1)$ .

Breadth-First Search on the other hand requires iterating through each movie with the first actor and then iterate over each cast member of each movie, pushing each onto a queue. In the worst case, we will have pushed every actor onto the node as well as traversed each actor-movie-actor connection, essentially every edge. This leads to  $O(|V|+|E|)$  in the worst case where  $|V|$  equals the count of all actors and  $|E|$  is the count of all actor-movie-actor connections. This is done on a year-by-year basis in our implementation to find the earliest year. After each year, visited actors must be “reset” to restart traversal for the next year, and thus incurs more overhead of  $O(|V|)$ . Granted it would be possible to perform BFS again for the “reset” which in the average case should be faster. Overall, this leads to a much slower query if all we care about is connectedness.