

We use the command time in the terminal when we run our actorconnection
(i.e.) time ./actorconnections ufind
time ./actorconnections bfs

Time To Find Actor Connection using BFS (average):

real: 13.257s
user: 12.103s
sys: 0.819s

Time To Find Actor Connection using UnionFind (average):

real: 2.527s
user: 0.518s
sys: 0.039s

From the result, we can see that by using UnionFind, we can greatly reduce the runtime for finding actors' connections. We believe that when the actors' path requires less movie connection(ie. two actors might be directly connected), the runtime for BFS and UnionFind should be at the same range. However, when it requires more movie path to link two actors, UnionFind data structure will significantly outperform BFS. The reason for this pattern to occur is that when using BFS to search for connection, we have to use a loop to loop through all node's neighbor and check which neighbor is the last actor in the path. However, if we use UnionFind data structure, when we union two actor together, all the actors along the path will point to the same root actor. As a result, by checking if two actors have the same root, we can conclude that whether they are connected in a path or not. Thus, it significantly save us a lot of time, since we don't have to loop through all the actor and actor's neighbors. Moreover, when adding connection between two actors, BFS requires us to do a double for loop to connect every actor with everyone else which cost $O(n^2)$ time. But, if we use UnionFind data structure, we don't need a double loop to connect every actor with everyone else, since all we do to connect those two actors is to make their prev point to the same root. Furthermore, for using BFS, every time we add one year worth of movies, we need to change the actor node's field (dist) back to its initial value for a successful find for BFS, since if we don't change it back, some actor might not be visit in the BFS. However, this process is not needed for union find, and thus save us a lot of time.