PA4 Report

Jiawei Li PID: A53226117
Ziyan Zhu PID:A53219779

September 6, 2017

Actor1	Actor2	Year	BFS Time	Union-Find Time
BACON, KEVIN (I)	HOUNSOU, DJIMON	1992	12.3867	0.191181
WILLIS, BRUCE	BACON, KEVIN (I)	1990	6.88592	0.189593
HOUNSON, DJIMON	50 CENT	2003	89.7084	0.191711
AARON, CAROLINE	WRIGHT, TOM (I)	1984	50.9536	0.196176
ZIEFF, RICK	HOWELL, KANIN	2000	50.9536	0.195585

Table 1: BFS vs Union-Find

From the running time in Table 1, we get the result that Union-find is way much faster than BFS, by calculating the average speed we found that Union-find is around 218.71 times faster than BFS search.

In general, the single BFS algorithm's time complexity is $O(|E| + |V|)$, and it is consistent for each search operation. In comparison, the worst-case running time of Union Find algorithm is $O(|V|)$ for the first time search operation, but it takes constant time for subsequent similar operations.

In our program, since the same actor pair appear 100 times, we need to run BFS and ufind algorithm 100 times for each year. Thus, the total time complexity for BFS in our program is $\#(year) \cdot O(|E| + |V|)$, while the total time complexity for ufind algorithm in our program is $O(|E| + |V| + (year))$.

For BFS, we actually apply it on a complete graph every time we search for the actor. Since we need to start with the earliest year, by each year, in the worst case we need to go through every node V and every edge E. In this case, the time complexity should be $\#(year) \cdot O(|E| + |V|)$.

While for union-find we have a very different searching method, we didn't construct the graph in the beginning, instead, we build our graph by iterating movie years in order. To start with, the graph will be created based on the movie and actors/actresses that appear in the earliest year, then whether our targets actors are already in this graph will be tested. If not, we will consider the actors/actresses in the movie in the next year into the graph accordingly and test the actor pair again. When building the graph, every node in a component is ensured to be pointed to the same node in this component as root. Thus, when we need to merge two component, only the roots of these two component are needed to be connected. Meanwhile, to search an actor pair, we only need to find the roots of these actors and compare whether the roots are the same. Since every time we call find(), we will generate all the node along the path from target to root pointing to root. When next time to search in this set, the time complexity will be constant $O(1)$.

Followed by these steps, after finishing building the graph, we only reach every edges and nodes once. Since we only need to add new edge and node every time we move to a new year, the time complexity of Union-find should be $O(|E| + |V| + \#(year))$.

Therefore, the larger data contained in the graph, the better performance ufind data structure has than BFS. Especially in this case, that 100 same actor pairs are needed to be searched, ufind saves plenty of time.