

//Name: Yuhang Lian

//Pid: A91018743

//Name: Bo Pang

//Pid: A91115104

FinalReport

For the BFS implementation, the run time collected by running 10 pairs is 37473560516. The average runtime is 37473560516. For the Union find algorithm, the total time taken for 10 pairs is 9423065069. The average case is 942306507. From the data we can tell that the Union Find Algorithm is 10 times faster than the BFS. It is because of the using of Path Compression. For multiple find of the same case, the union find will update the target's parent to sentinel node after the first time we run the algorithm. So for the following times, the time taken to track down required data will take less and less time. For the BFS, because it needs to keep go down the tree and look for the required data, it will take a considerably longer time compare to Ufind.

Which implementation is better and by how much?

By comparing two results, Union find is generally better than BFS because it takes 10 times less time than BFS does.

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

Because of the optimization brought by Path compression, the Union find is able to optimize the run time after each search. Therefore, if the number of data need to be located is large, Union find is better than bfs.

What arguments can you provide to support your observations?

The results from the tests described above can fully prove my observation.