The social network should be represented as a graph using adjacent lists for each user. I choose this representation because it has several advantages, such as reducing storing information about non-friendship links. This graph tends not to be complete for a high number of users (e.g. one user is a friend with anybody in the network). The algorithm that stands behind my solution to this problem is Bidirectional Search, and the alternatives are BFS and DFS. I prefer to implement this algorithm because it is faster than the alternative searches.

Test cases:

Structure of the file

List of users’ names

UserA’s and UserB’s name (should appear on the first line)

The edges between users according to their position on the first line.

1.

A B C D E F G H I Z

A Z

0 1

1 3

1 2

2 3

3 5

5 6

6 7

5 7

2 4

4 8

No chain between A and Z

2.

A B C D E F G H I Z

A I

0 1

1 3

1 2

2 3

3 5

5 6

6 7

5 7

2 4

4 8

Take one chain

3.

A B C D E F G H I Z

A B

0 1

1 3

1 2

2 3

3 5

5 6

6 7

5 7

2 4

4 8

Very close friends

4.

A B C D E F G H I Z

B F

0 1

1 3

1 2

2 3

3 5

5 6

6 7

5 7

2 4

4 8

Take the shortest chain