

A  
Project Report  
On  
**“It’s a Small World”**  
Prepared by

885185645 | Devashri Manepatil | [devashri@csu.fullerton.edu](mailto:devashri@csu.fullerton.edu)  
885209536 | Apoorva Machale | [apoorvamachale@csu.fullerton.edu](mailto:apoorvamachale@csu.fullerton.edu)  
886679687 | Sumit Bishnoi | [sumitbishnoi@csu.fullerton.edu](mailto:sumitbishnoi@csu.fullerton.edu)

**A Report Submitted to**  
Prof. Doina Bein  
Department of Computer Science  
College of Engineering and Computer Science (ECS)  
California State University, Fullerton (CSUF)  
for the Fulfillment of the Requirements for the  
**CPSC 535: Advanced Algorithms – Project 2**



California State University, Fullerton  
CA - 92831  
December 2022

# Chapter 1

## Summary

It's a Small World algorithm based on the popular 2019 game "Six Degrees of Kevin Bacon". The goal of this algorithm is to find the shortest path between two cast members from different movies. And the casts of various films will be saved in a set in string format. We have written the code in python language. The input text file will consist of the number of casts and casts. The output gives a set of casts if the shortest distance is 1 or 2 else we just print the output saying that the connection distance is more than 2.

As a result, this algorithm will assist us in determining the relationship between movie casts.

## Chapter 2

### Problem Statement

Given a spin-off of the popular game “Six Degrees of Kevin Bacon” invented in 2019. Let’s say we have casts for two movies, and we want to find out what is the shortest connection between the two casts.

We assume that the name of each actor is a string, thus a cast is a set of strings, ordered in alphabetical order. But it does not matter whether they are listed alphabetically or not, but for simplicity, let’s list them in alphabetical order.

The input will be a positive integer  $n > 2$ , and a list of  $n$  casts from which the first two sets are more significant,  $CAST[0]$  and  $CAST[1]$ . If the two casts  $CAST[0]$  and  $CAST[1]$  have at least one string in common, then the shortest connection is 1. If the two casts  $CAST[0]$  and  $CAST[1]$  do not have any string in common, then look for another cast in the list of  $n$  casts, let’s called it  $tempCast$ , such that  $CAST[0]$  and  $tempCast$  have a string in common, and  $CAST[1]$  and  $tempCast$  have a string common, then the shortest connection is 2. Else the shortest connection is greater than 2 or there is no connection.

## Chapter 3

### Pseudocode

**Step 1:** We take input from a txt file which contains the number of inputs(n) and the list of movie casts, i.e casts=[].

**Step 2:** We start reading the first line of the file by `number_of_casts = str(f.readline())[2:-4]` and convert `n = int(number_of_casts)`

**Step 3:** Start reading each line using loop read the set of casts and append in the list cast i.e. `casts.append(set(line))`

**Step 4:** If `n > 2`, then we check if we have common casts in `cast0` and `cast1`, i.e. `cast[0].intersection(cast[1])` and store it in temporary variable, `z`.

**Step 5:** We then apply for loop from `(2,n)` to compare `cast[0]` and `cast[1]` with `cast[i]`, i.e. `cast[0].intersection(cast[i])` and `cast[1].intersection(cast[i])`, which helps to check if there is any common casts present. Store length of common casts for `cast0` and `cast1` present in `matching_cast0_len` and `matching_cast1_len` respectively.

**Step 6:** If `(matching_cast0_len == 0 and matching_cast1_len == 1)`; then the counter is incremented, i.e `counter = counter + 1` and we store the `tempCast = cast[i]`.

**Step 7:** Similarly, when we run the loop again and when we find a common cast then we will again increment counter, but if `counter > 2` then break the loop.

**Step 8:** If our `len(z)` is 1 then we say that the shortest connection is 1 and display the common cast.

**Step 9:** If `counter > 2` or `counter == 0` then we say that the shortest connection is more than 2 or no connection.

**Step 10:** If `counter == 2`, then the shortest connection is 2 and print the `tempCast`.

## Chapter 4

# Project Setup Guidelines

### **Method-1: If you have any IDE (VSCode/ PyCharm)**

1. Download the file algorithm\_2.py and input.txt
2. Install the python compiler on the IDE
3. Open the file in IDE and click on run
4. You can enter the input details and get the desired output in Terminal of the IDE

### **Method-2: If you are using Terminal (Command Prompt)**

1. Download the file algorithm\_2.py and input.txt
2. Open command prompt
3. Go to the location of downloaded file
4. Run the file using command: `python algorithm_2.py`

## Chapter 5

### Screenshots

#### Example 1

**Input:** We are given value of  $n=6$  and casts list which is:

CAST[0] = { "Carrie-Anne Moss", "Gloria Foster", "Hugo Weaving", "Joe Pantoliano", "Keanu Reeves", "Laurence Fishburne", "Marcus Chong" }

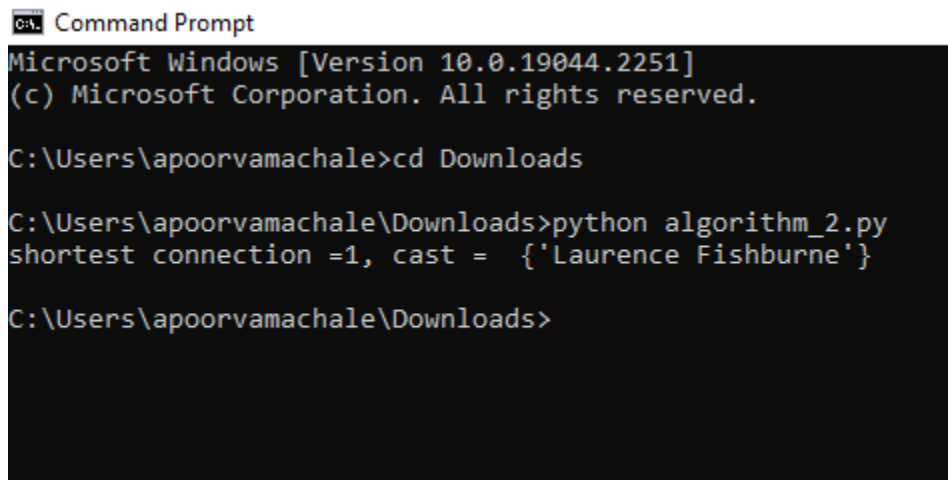
CAST[1] = { "Andre Braugher", "Beau Garrett", "Chris Evans", "Doug Jones", "Ioan Gruffudd", "Jessica Alba", "Julian McMahon", "Kerry Washington", "Laurence Fishburne", "Michael Chiklis" }

CAST[2] = { "Ewan McGregor", "Ian McDiarmid", "Jake Lloyd", "Liam Neeson", "Natalie Portman" }

CAST[3] = { "Geoffrey Rush", "Jack Davenport", "Johnny Depp", "Jonathan Pryce", "Keira Knightley", "Orlando Bloom" }

CAST[4] = { "Angela Bassett", "Chadwick Boseman", "Danai Gurira", "Daniel Kaluuya", "Forest Whitaker", "Letitia Wright", "Lupita Nyong'o", "Martin Freeman", "Michael B. Jordan", "Sterling K. Brown", "Winston Duke" }

CAST[5] = { "Andrew Borba", "Anne Hathaway", "Bill Irwin", "Casey Affleck", "Collette Wolfe", "David Oyelowo", "Francis X. McCarthy", "Jessica Chastain", "John Lithgow", "Matthew McConaughey", "Michael Caine", "Wes Bentley", "William Devane" }



```
CA: Command Prompt
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\apoorvamac hale>cd Downloads

C:\Users\apoorvamac hale\Downloads>python algorithm_2.py
shortest connection =1, cast = {'Laurence Fishburne'}

C:\Users\apoorvamac hale\Downloads>
```

## **Example 2**

**Input:** We are given value of  $n=7$  and casts list which is:

CAST[0] = { "Carrie-Anne Moss", "Gloria Foster", "Hugo Weaving", "Joe Pantoliano", "Keanu Reeves", "Laurence Fishburne", "Marcus Chong" }

CAST[1] = { "Andrew Borba", "Anne Hathaway", "Bill Irwin", "Casey Affleck", "Collette Wolfe", "David Oyelowo", "Francis X. McCarthy", "Jessica Chastain", "John Lithgow", "Matthew McConaughey", "Michael Caine", "Wes Bentley", "William Devane" }

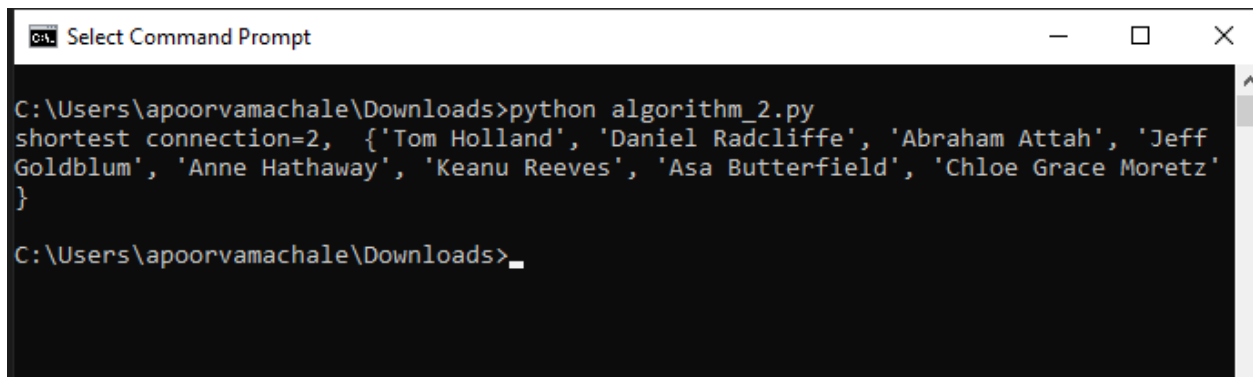
CAST[2] = { "Geoffrey Rush", "Jack Davenport", "Johnny Depp", "Jonathan Pryce", "Keira Knightley", "Orlando Bloom" }

CAST[3] = { "Angela Bassett", "Chadwick Boseman", "Danai Gurira", "Daniel Kaluuya", "Forest Whitaker", "Letitia Wright", "Lupita Nyong'o", "Martin Freeman", "Michael B. Jordan", "Sterling K. Brown", "Winston Duke" }

CAST[4] = { "Abraham Attah", "Asa Butterfield", "Anne Hathaway", "Chloe Grace Moretz", "Daniel Radcliffe", "Jeff Goldblum", "Keanu Reeves", "Tom Holland" }

CAST[5] = { "Andre Braugher", "Beau Garrett", "Chris Evans", "Doug Jones", "Ioan Gruffudd", "Jessica Alba", "Julian McMahon", "Kerry Washington", "Laurence Fishburne", "Michael Chiklis" }

CAST[6] = { "Ewan McGregor", "Ian McDiarmid", "Jake Lloyd", "Liam Neeson", "Natalie Portman" }



```
C:\Users\apoorvamachale\Downloads>python algorithm_2.py
shortest connection=2, {'Tom Holland', 'Daniel Radcliffe', 'Abraham Attah', 'Jeff
Goldblum', 'Anne Hathaway', 'Keanu Reeves', 'Asa Butterfield', 'Chloe Grace Moretz'
}
C:\Users\apoorvamachale\Downloads>_
```

### **Example 3**

**Input:** We are given value of  $n=7$  and casts list which is:

CAST[0] = { "Ewan McGregor", "Ian McDiarmid", "Jake Lloyd", "Liam Neeson", "Natalie Portman" }

CAST[1] = { "Andrew Borba", "Anne Hathaway", "Bill Irwin", "Casey Affleck", "Collette Wolfe", "David Oyelowo", "Francis X. McCarthy", "Jessica Chastain", "John Lithgow", "Matthew McConaughey", "Michael Caine", "Wes Bentley", "William Devane" }

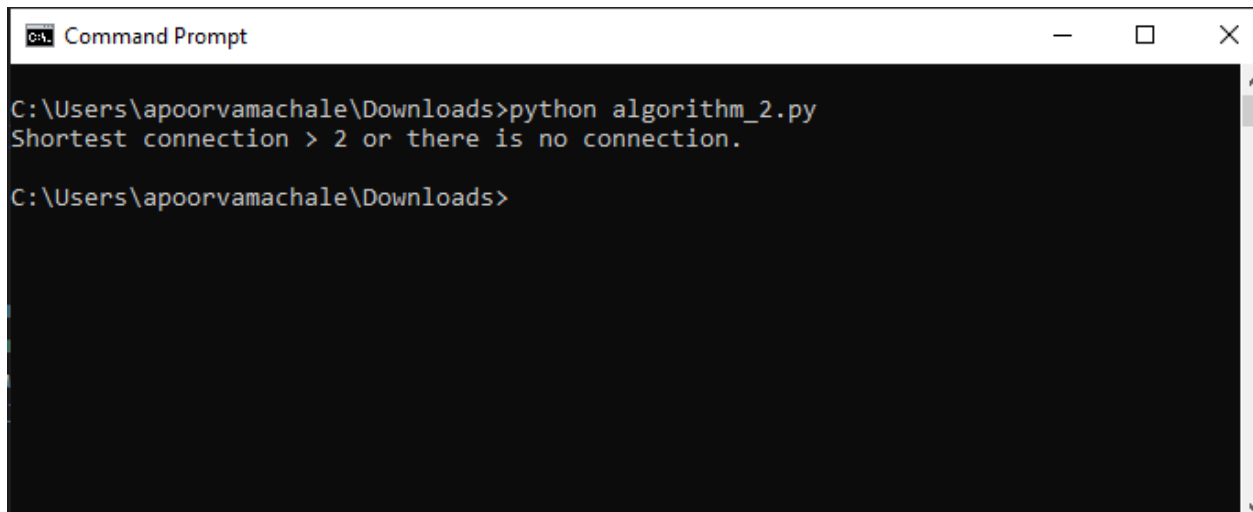
CAST[2] = { "Geoffrey Rush", "Jack Davenport", "Johnny Depp", "Jonathan Pryce", "Keira Knightley", "Orlando Bloom" }

CAST[3] = { "Angela Bassett", "Chadwick Boseman", "Danai Gurira", "Daniel Kaluuya", "Forest Whitaker", "Letitia Wright", "Lupita Nyong'o", "Martin Freeman", "Michael B. Jordan", "Sterling K. Brown", "Winston Duke" }

CAST[4] = { "Abraham Attah", "Asa Butterfield", "Anne Hathaway", "Chloe Grace Moretz", "Daniel Radcliffe", "Jeff Goldblum", "Keanu Reeves", "Tom Holland" }

CAST[5] = { "Andre Braugher", "Beau Garrett", "Chris Evans", "Doug Jones", "Ioan Gruffudd", "Jessica Alba", "Julian McMahon", "Kerry Washington", "Laurence Fishburne", "Michael Chiklis" }

CAST[6] = { "Carrie-Anne Moss", "Gloria Foster", "Hugo Weaving", "Joe Pantoliano", "Keanu Reeves", "Laurence Fishburne", "Marcus Chong" }



```
C:\Users\apoorvamachale\Downloads>python algorithm_2.py
Shortest connection > 2 or there is no connection.

C:\Users\apoorvamachale\Downloads>
```



## Chapter 6

### Group Member Details

README.md



### 535-project2

It's a small world

Summary: The It's a Small World algorithm is based on the popular 2019 game "Six Degrees of Kevin Bacon." The goal of this algorithm is to find the shortest path between two casts from different movies. And the casts of various films will be saved in a list in string format.

We will be given a positive integer greater than 2, and the first two movie casts will be very significant because we will find the shortest connection between these, and if we find at least one, we can say we have one shortest connection. If there is no connection between the first two movie casts, we will look for another cast from the list of casts. If we discover a link between them, the shortest distance will be two. Otherwise, we return the shortest connection that is greater than two or no connection.

As a result, this algorithm will assist us in determining the relationship between movie casts.

Group members:

Devashri Manepatil: [devashri@csu.fullerton.edu](mailto:devashri@csu.fullerton.edu)

Apoorva Machale: [apoorvamachale@csu.fullerton.edu](mailto:apoorvamachale@csu.fullerton.edu)

Sumit Bishnoi: [sumitbishnoi@csu.fullerton.edu](mailto:sumitbishnoi@csu.fullerton.edu)