# Report

SID: 500508958

## Section 1 - Testing

Q1. Three good reasons to write test cases are:

1) Ensures that the program works as intended, accounting for faults and errors
2) Helps the developer of the code identify any bugs/errors within the code
3) Also ensures that contingency's are put into place in case an error does occur

Q2. With testing, mocks are used to simulate live conditions and behaviour of a program. Essentially this allows for the developer to make sure that certain areas (classes, modules etc.) of a program execute as expected. Mocks can be anything within the system (objects, classes etc.)

Two advantages of mocking include:

1) Test the system more thoroughly by changing the type of mock in use
2) Saves the developer time and effort of having to use external dependencies

Two disadvantages of mocking include:

1) It can be time consuming in writing and managing the mock types used.
2) Mocking doesn't necessarily test the interaction between classes, modules etc., only what the classes, modules return or execute.

In summary, mocks should generally be used in unit testing, rather than end-to-end tests.

Q3. A real life example of where insufficient code testing led to problems was the situation in mid January 2016 the Nest thermostat (AC product) was affected by a software bug in firmware update, leaving users in the cold. According to Computer World, the bug had forced the device's battery to drain, leaving users unable to control the temperature within their homes. This is an example of insufficient code testing because had the developers tested the software before rolling it out to the public, then the device's battery would be unaffected.

# Section 2 - Solver

Q1. The strengths of BFS include:

1) Scans through items in a program (or grid) level by level, so it will or will not find a solution (depending on the structure)
2) Always finds the shortest path from an object to its neighbours.

The weaknesses of BFS include:

1) Method occupies more memory (has to go through each node as such)
2) Slower and takes longer to produce a given solution than the DFS algorithm

BFS would be used in solving shorter structures, given that it takes up a lot of space

Q2. The strengths of DFS include:

1) Consumes less memory and time when solving a structure
2) Is able to find solutions in structures that are longer and contain more elements.

The weaknesses of DFS include:

1) Does not always find the most optimal solution, given that it moves onto the next node and removes the most recent node visited
2) DFS structure is determined by how many paths there are, so it may search a useless path

DFS would be used in solving more complex structures that contain different types of cells with different functionality.

Q3) If the ending cell is closer to the starting cell, DFS isn't necessarily guaranteed to be faster than BFS, depending on the cells that are between the top and bottom row of the map.

# References

*What is Mocking ?* 2010, StackOverflow, viewed 29th May, https://stackoverflow.com/questions/2665812/what-is-mocking

Narendra, arvind pdmn, *Mock Testing*, Devopedia, viewed 29th May, https://devopedia.org/mock-testing

*To mock or not to mock?* 2008, Microsoft, viewed 29th May, https://docs.microsoft.com/en-gb/archive/blogs/peaceofmind/to-mock-or-not-to-mock

*Top software failures in recent history*, COMPUTERWORLD 2020, viewed 29th May, https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html#slide12