Section 1

1. Reasons for test cases
    a. Aids debugging of programs while modifying and developing them
    b. Allows more focused approach to program development e.g. Test driven programming
    c. Considering all positive, negative and edge cases allows identification of potential gaps within the program

2. Mocks are objects that simulate the behaviour of objects which the unit may be dependent on. They are generally used for unit testing. (arvindpdmn, 2020)

Advantage	Disadvantage
• Isolates code being tested, does not focus on external dependencies • External dependencies do not need to be fully developed to test code	• Can add unnecessary complexity to code when over used • Good understanding of dependencies is required to write mock which accurately represent normal behaviour

3. Ola is India's largest taxi aggregator, their mobile app had software bugs that allowed people with basic programming knowledge to have unlimited free taxi rides at the expense of Ola and other riders' wallets (Aggarwal and Murali, 2020). Ola did not sufficiently test their encryption for user data and the code responsible for their first-time user scheme.

Section 2

1. BFS algorithm traverses the cells of the grid level by level, that is It traverses all the 1 move paths then all the 2 move paths and so on. BFS algorithms are suited for use on grids with short solution paths (due to run-time) or when the shortest solution is required.

Advantages	Disadvantage
• Finds shortest possible path	• Has memory constraints due to it storing all nodes of current level • The further the end cell is from the starting cell the greater the amount of paths that the BFS algorithm will traverse, this increase is usually not linear and can lead to very long run-time

2. DFS algorithm traverses the full depth of a path before backtracking and finding another possible path. DFS algorithm is suitable for use on grids with long solution paths where the shortest solution is not required

Advantages	Disadvantage
• Uses less memory than BFS • Faster at finding distant end cells	• Path is not guaranteed to be the shortest • May search depth of paths with no solutions

3. If the ending cell is close to the starting cells, DFS is not always faster than BFS. For example In the grid below the DFS could search the depths of the dead end and be slower than BFS due to more possible paths needing to be traversed before finding solution.

*	X	*	*	*	*	*
*						*
*	Y	*	*	*	*	*

4. The water and fire cell features of the game creates situations where it is required to revisit some cells. An example of such a situation:

*	X	*	*	*
*				*
*		*		*
*		*		*
*		*		*
*		*		*
*	W	*	F	*
*	*	*	Y	*

The solution to this grid required the player to step on the water cell to get a water bucket then extinguish the fire blocking the end, but the water cell is at a dead end meaning the player must revisit some cells to reach the end cell. This prevents the normal use of a list of visited cells which is used in BFS and DFS algorithm. Such a list can still be implemented in the solver for this game if the list is emptied whenever a water bucket it gained.

Bibliography

Aggarwal, V. and Murali, M., 2020. *Taxi Aggregator Ola Hit By Tech Glitches That Allow Free Rides*. [online] The Economic Times. Available at: <https://economictimes.indiatimes.com/small-biz/startups/taxi-aggregator-ola-hit-by-tech-glitches-that-allow-free-rides/articleshow/46629010.cms> [Accessed 29 May 2020].

arvindpdmn, n., 2020. *Mock Testing*. [online] Devopedia. Available at: <https://devopedia.org/mock-testing> [Accessed 28 May 2020].