WRITE UP

A)

As you increase the number of junctions in the maze, the number of possibilities rises exponentially. If we add a new junction, it means we have to check that new junction's connections, and how the other junctions connect to the new junction as well, adding a layer to check. This results in the runtime of our program to go up. As stated previously in the pseudocode, 2^n is related to the amount of possible paths there will be, vaguely. If there are 4 junctions that may or may not connect to one another, 2^4 is roughly the amount of paths there are.

B)

Coding wise, this problem is not hard to code. Computationally, it is very hard to find all the paths, especially if the number of junctions rose. With not even half the alphabet used, one of the sample files used still had around 300 possible paths to be found. In order to traverse the maze and find all the possible paths, we needed to check every possible path and see if it lead to Z. Computationally, this is very time consuming, but logically not hard to understand.