**Maze APPROACH**

I am going to implement Depth First Search (DFS) in parallel processes to find the path in the maze. The reasons for opting DFS over Breadth First Search (BFS) are:

* DFS is faster than BFS.
* Even using the already implemented single threaded BFS and DFS, we can see that DFS gets the path faster than BFS.
* DFS uses memory more efficiently than BFS
* The main reason is that BFS returns shortest path whereas DFS return just the path

**Implementing DFS as multithreaded solution:**

* We use DFS here in this solution on every choice to get the path.
* So this tasks should be shared among threads to make the solution faster
* The first thing to do is to implement DFS as a callable
* This allows us to capture all the returned paths from all parallelly running threads using DFS
* To capture results from all the DFS callables we need Futures.
* So add all the results to a Future list
* To invoke the future objects we need executor service.