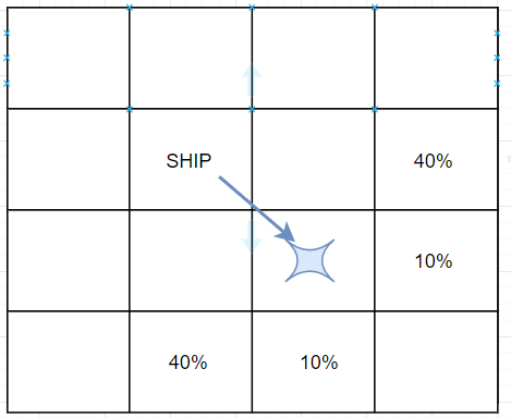
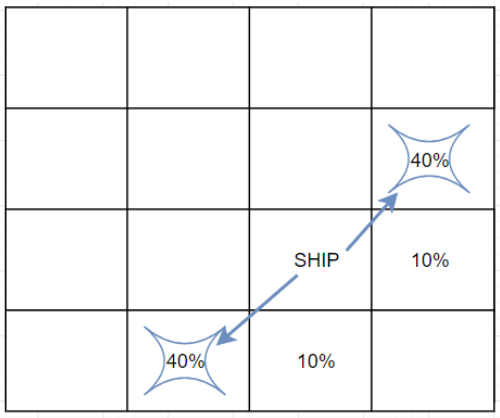
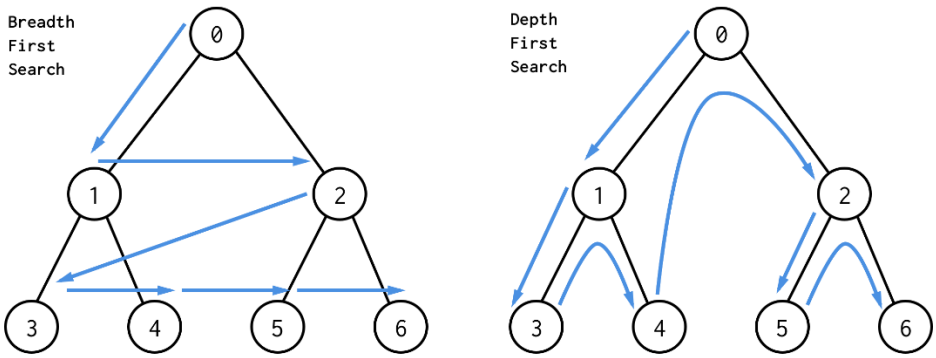
Task 3: Discussion

In regards to task 1 I assumed the solution to the problem was far from the starting position, so in order to execute a less expensive search and find the answer more quickly, I opt to perform a “Depth First Search Algorithm”, this algorithm searches from a starting position all possible moves and proceeds to execute the same action on the first branch (in a tree format), in case of no more possible moves for that branch, it goes back one move and continues to search from that point forward until the solution is reached. Even though this algorithm is best to find solutions more quickly, the number of moves it takes to do so is most of the times affected.

In the scenario of the ship finding the whale, in order to make sure the ship executes the least amount of moves (and thus finding the whale more quickly), it should be applied the “Breadth First Search Algorithm”, this algorithm searches every possible move for each level of the tree until the goal is found. The initial state of the scenario previously mentioned would be the position of ship, the final goal the position of the whale (in case of the whale moving the goal would also need to be changed for each interaction) and the successors would be the possible moves of the ship on the grid from current position. Since the location of the whale is unknown, it is necessary to calculate the possible positions of the whale in the grid given the distance “d” between them, and also implement a point system that forces the ship to only take moves that reduces or maintains the distance “d”, in case of whale not moving it is essential the ship doesn’t take a position he already has been to (avoiding loops), otherwise simply take the move with highest probability (in case of more than one position with same highest probability, if distance is bigger than one and there’s a middle position between those high probability positions move there, otherwise move randomly between them).







(Practice.geeksforgeeks.org, 2019)

In the light of the minesweeper problem, a conservative search algorithm is based on the premise of making safe plays, which most consist of working around the numbers and flagging the mines when the location is certain. On the other hand, in order to accelerate the identification of the mines, it is necessary to have a more aggressive behavior (as said in task 3 introduction), this can be accomplished by taking risks, for example selecting a random position of grid is hopes of not stepping on a mine and open a big area.

In order to apply the ship and whale algorithm previously described into the minesweeper problem the start state needs to be updated to a counter with the value 0, the goal state as the total number of mines, and the successors the positions on the grid that aren't mines. The successors list will be updated with every interaction by removing the positions already selected and adding new possible moves. Each move can trigger one of two responses, flag a mine (increasing the counter) or clicking on a safe position, a point systems should be implemented in order to prioritize flagging the mines and the moves that will give away a mine position. If there is no possible safe moves left a probability system should be implemented to select the next move according to it's probability of not being a mine.

(Let's Learn How to Play Minesweeper (EP2) Probability of a Mine (15 Jun 2014))

References:

- “Let's Learn How to Play Minesweeper (EP2) Probability of a Mine” (15 Jun 2014), YouTube video, added by “DeRockProject & the Attack of the Really Long Channel Name”: https://www.youtube.com/watch?v=_qINGZvIR-0
- Practice.geeksforgeeks.org. (2019). Difference between BFS abd DFS. | Practice | GeeksforGeeks. [online] Available at: https://practice.geeksforgeeks.org/problems/difference-between-bfs-abd-dfs [Accessed 30 Apr. 2019].