

Project Brief : Implementation of the (2n+1) tiles problem using search methods.

Problem Statement :

\*Given 2N Tiles and 1 Empty Space.The Goal is to arrange the tiles in the order such that one red tile should at the last position and all the green tiles are arranged to the left of the Red tiles.The cost of each move is the distance between the tile and the empty space where it moves.There are at most 2N legal moves are possible.

SEARCH ALGORITHMS USED:

BREADTH FIRST SEARCH:

\* The algorithm which expands all nodes at a time and start searching in breadthwise.the algorithm moves to the next level only if it completes the search on the previous level nodes.

DEPTH FIRST SEARCH:

\*This search technique expands in depth wise manner to the depth of the level that is defined and start searching recursively from there.it is a type of non-heuristic search since no pathcost and heuristic values are involved.

BEST FIRST SEARCH:

\*This search technique takes the best weighted score that is the heuristic value and compares it with all of its successors and choose the least heuristic valued node and goes on and traces back.

A \* SEARCH:

\*This search technique selects the path based on both the heuristic value and the path cost.it uses the priority queue to do this operation.

HILL CLIMBING:

\*This search is similar to the best first search , it takes the heuristic value and moves according and won’t traverse back.

KEY CONCEPTS IMPLEMENTED :

► Dictionary,Trees

► String Handling

► List Data Structures.

► Functions

► Iterations and Loops

► Control and Conditional Statements

► Elementary Operators.

BEST ALGORITHM IN THIS PROBLEM:

* A Star is the best performing algorithm in this problem since,it uses the both pathcost and heuristic value for the movement it is performing best when compared to all.It gives only the precised the movements as soon as possible.