Title: Developing best first search(BFS) and A* algorithm for real world problems

EX. NO : 05 Name: Parth Langalia

DATE : 20-02-2023 **Reg No. :** RA2011033010033

AIM :

Implementation and Analysis of BFS and A* Search for real world problems.

PSEUDO CODE :

Algorithm for Best First Search:

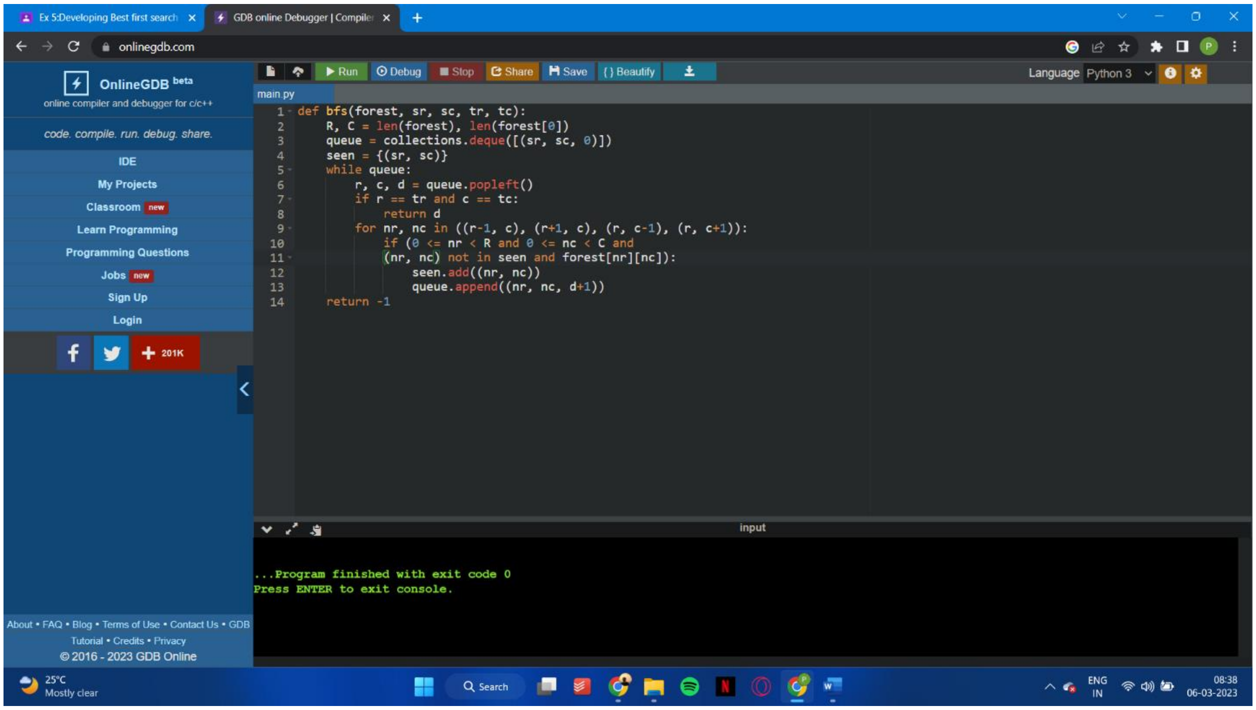
1. Perform best-first-search, processing nodes (grid positions) in a queue.
2. Seen keeps track of nodes that have already been added to the queue at some point.
3. Those nodes will be already processed or are in the queue awaiting processing.
4. For each node that is next to be processed, look at it's neighbors. If they are in the forest (grid), they haven't been enqueued, and they aren't an obstacle, we will enqueue that neighbor.
5. Keep a side count of the distance travelled for each node. If the node we are processing is our destination 'target' (tr, tc), we'll return the answer.

Algorithm for A* search:

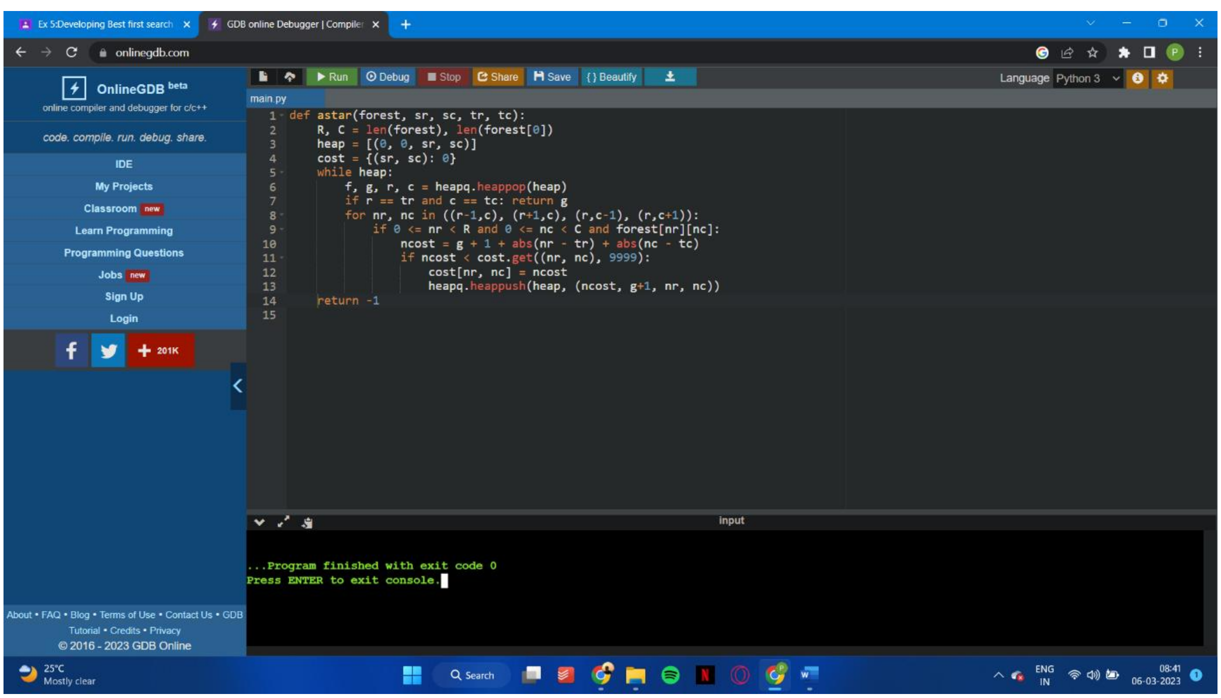
1. The A* star algorithm is another path-finding algorithm
2. For every node at position (r, c), have some estimated cost node.f = node.g + node.h
3. node.g is the actual distance from (sr, sc) to (r, c).
4. node.h is our heuristic (guess) of the distance from (r, c) to (tr, tc).
5. The taxicab distance, node.h = abs(r-tr) + abs(c-tc).
6. Keep a priority queue to decide what node to search in (expand) next.

PROGRAM(with OUTPUT) :

BFS Search:



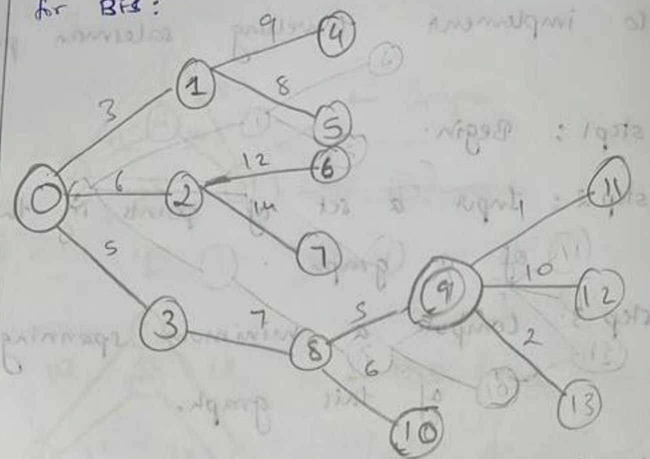
A* Search:



Manual Calculations :

Manual calculations / Input & Output :

for BFS :



Input : start : 0

Destination : 9

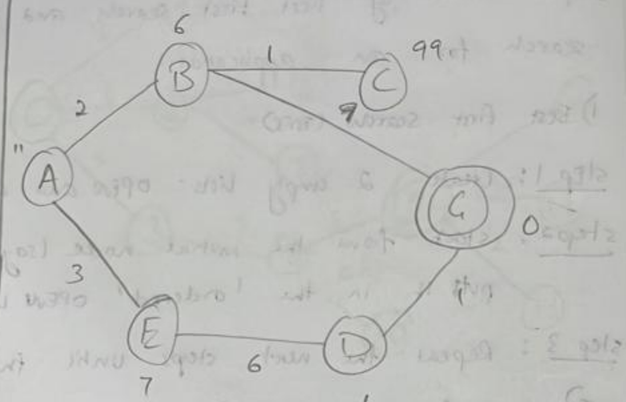
output :

0 → (1, 3, 2) (ascending order)

0 → (5, 4, 8, 2) ← traverse further into node 1 and node 3.

∴ Path = (0, 1, 3, 2, 8, 9)

for A* algorithm :



Input : start point : A

Destination point : G

Output :
starting at A,

AB is better than AE as

AE > AB.

(A*) o/p verified

∴ AB is selected.

1) ABC rejected as non-terminating node.

2) ABG = 11 units.

BUT

AEDG is shorter path with 10 units.

∴ (A, E, D, G) is the shortest path.

RESULT :

The cutting off of tree problem for a golf event(real world problem) successfully solved with 2 different approaches : Best first search and A* search algorithm.