AI

VARIN KALA
2022561

ASSIGNMENT-1

Date ..../..../....

Q1)

a) A^* Search Algorithm:

$$f(n) = g(n) + h(n)$$

$g(n)$ - Path cost from node 'S' to 'n'
$h(n)$ - heuristic (h-values of nodes)

① $f(S) = 0+8 = 8$ Frontier = {A, B, C}

② $f(A) = g(A) + h(A) = 3+2 = 5$
$f(B) = 1+1 = 2$
$f(C) = 5+8 = 13$

'B' is traversed; Frontier = {A, C, D, F, G3}

③ $f(D) = 5+4 = 9$
$f(F) = 3+3 = 6$
$f(G3) = 13+0 = 13$
& $f(A) = 5$, $f(C) = 13$

'A' is traversed; frontier = {C, D, F, G3, G1}

④ $f(G1) = 13+0 = 13$
$f(C) = 13$, $f(D) = 9$, $f(F) = 6$, $f(G3) = 13$

'F' is traversed; frontier = {C, D, G3, G1}

⑤ $f(D) = 4+4 = 8$
$f(C) = 13$, $f(G3) = 13$, $f(G1) = 13$

'D' is traversed; Frontier = {C, G3, G1, E, G2}

Spiral

Date ..../..../........

(6) $f(E) = 7+1 = 8$
$f(G2) = 9+0 = 9$
$f(C) = 13$, $f(G3) = 13$, $f(G1) = 13$

'E' is traversed; Frontier = {C, G3, G1, G2, ~~[illegible]~~}

(7) $f(G1) = 8+0 = 8$
$f(C) = 13$, $f(G3) = 13$, $f(G1) = 13$, $f(G2) = 9$

'G1' is traversed;
We have found the goal node

∴ Path = S → B → F → D → E → G1
Order of Nodes Expanded
= S → B → A → F → D → E → G1

b) Uniform Cost Search
$f(n) = g(n)$

$g(n)$ - Path Cost from node 'S' to 'n'

(1) $f(S) = 0$ Frontier = {A, B, C}

(2) $f(A) = 3$, $f(B) = 1$, $f(C) = 5$
'B' is traversed; Frontier = {A, C, D, F, G3}

(3) $f(D) = 5$, $f(F) = 3$, $f(G3) = 13$, $f(A) = 3$, $f(C) = 5$
'A' is traversed; Frontier = {C, D, F, G3, G1}

Spiral

Date ..../..../........

④ $f(G1) = 13, f(C) = 5, f(D) = 5, f(F) = 3, f(G3) = 13$
'F' is traversed; frontier = {C, D, G1, G3}

⑤ $f(D) = 4, f(C) = 5, f(G1) = 13, f(G3) = 13$
'D' is traversed; frontier = {C, G1, G3, E, G2}
⑥ $f(E) = 6, f(G2) = 9, f(C) = 5; f(G1) = 13, f(G3) = 13$
'C' is traversed; Frontier = {G1, G3, E, G2, }

⑦ 'E' is traversed; Frontier = {G1, G3, G2}

⑧ $f(G1) = 8, f(G2) = 9, f(G3) = 13$

'G1' is traversed;
We have found the goal node

∴ Path = S → B → F → D → E → G1

∴ Order of Nodes Expanded
= S → B → A → F → D → C → E → G1

c) Iterative deepening A* Search

① $f(S) = 0$, $\boxed{\text{Threshold} = h(S) = 8}$ Frontier = {A, B, C}

② $f(A) = 5, f(B) = 2, f(C) = 13$ & ~~frontier =~~

'B' is traversed; frontier = {A, D, F, G3}

③ $f(A) = 5$, $f(D) = 8$, $f(G_3) = 13$, $f(F) = 6$

'A' is traversed; Frontier = {D, F, G1}

④ $f(G_1) = 13$, $f(D) = 8$, $f(F) = 6$

'F' is traversed; Frontier = {D}

⑤ $f(D) = 8$

'D' is traversed; Frontier = {E, G2}

⑥ $f(E) = 7$, $f(G_2) = 9$

'E' is traversed; Frontier = {G1}

⑦ $f(G_1) = 8$

'G1' is traversed;

We have found the goal node

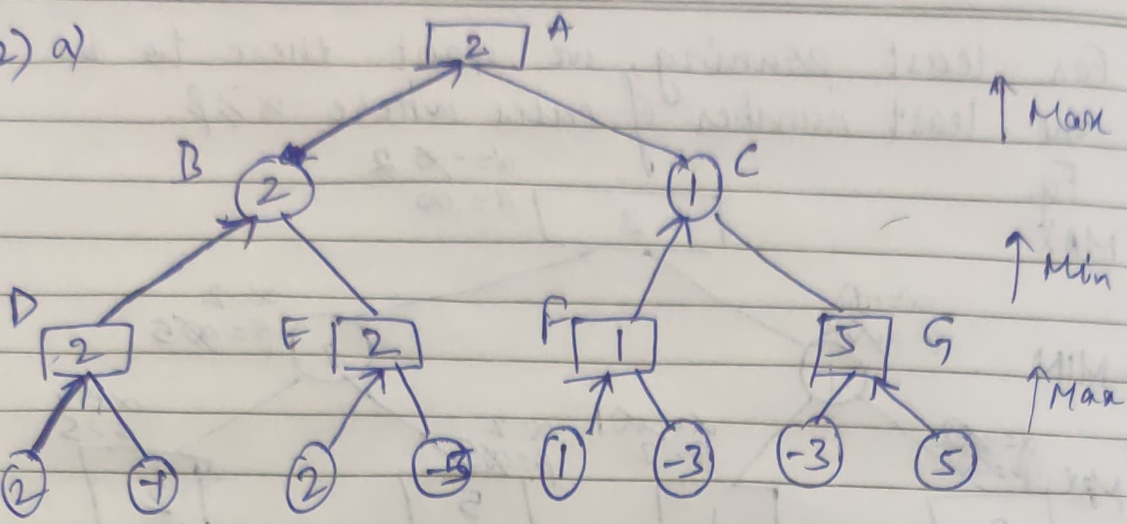
Path = $S \rightarrow B \rightarrow F \rightarrow D \rightarrow E \rightarrow G_1$

∴ Order of Nodes Expanded

$= S \rightarrow B \rightarrow A \rightarrow F \rightarrow D \rightarrow E \rightarrow G_1$

Date ..../..../....

Q2) a)

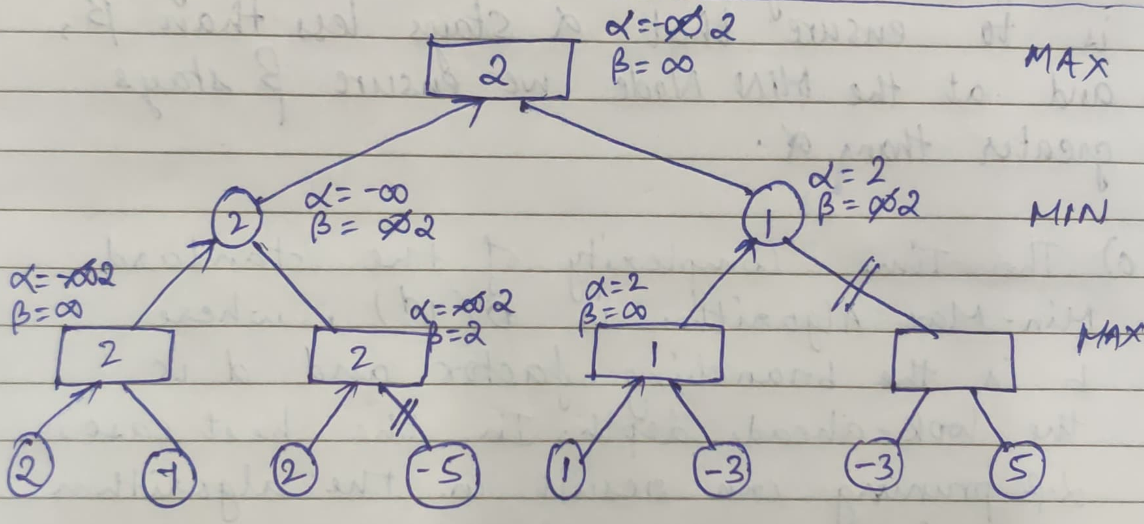


Arrows → Indicate Best Moves

Alpha-Beta Pruning:

$\alpha \rightarrow$ Max $\beta \rightarrow$ Min

Prune when: $\alpha \geq \beta$



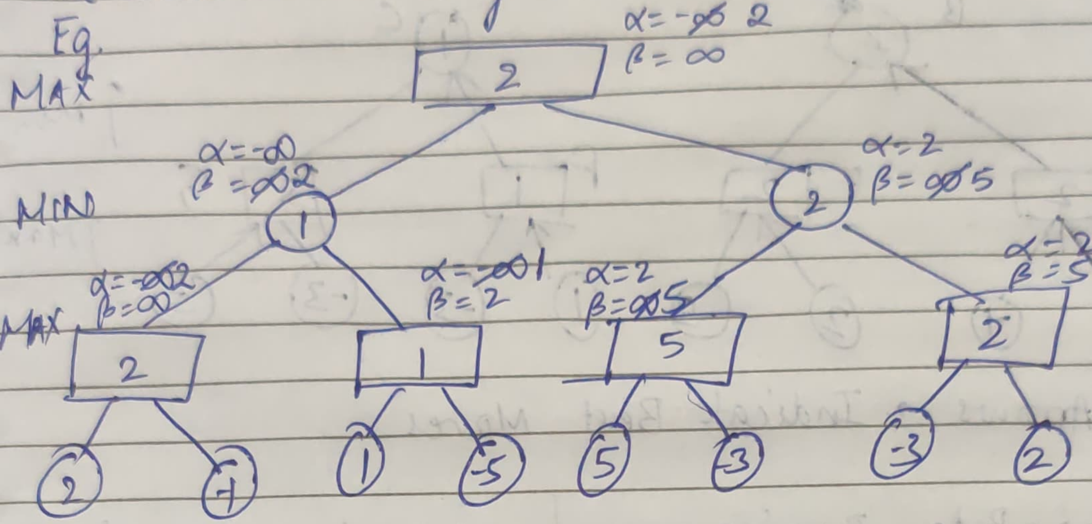
b) For Maximum ~~In Max~~ Pruning, we want to ensure early cuts or prunes. ~~We can do this by arranging the most extreme values as early as possible.~~ We have to maximize the chances of $\alpha \geq \beta$. One such example is the one given in the question.

Spiral

Date ..../..../........

For least pruning, we want there to be the least number of cases where $\alpha \geq \beta$.

Eg.



Here, at every ~~max~~ MAX Node, our effort is to ensure that α stays less than β, and at the MIN Node we ensure β stays greater than α.

c) The Time Complexity of the standard Min-Max Algorithm is $O(b^d)$, where b is the branching factor and d is the look-ahead depth. In the best case, α-β pruning can result in the algorithm searching only half the tree. Hence, the effective depth becomes $d/2$ and the Time Complexity becomes $O(b^{d/2})$

Q3)
b) The path obtained from running the 2 uninformed algorithms may be the same but not always. The Iterative Deepening Search algorithm works on the idea of DFS, while the Bidirectional Breadth-First Search algorithm works on the idea of BFS, and depending on the precedence to which set of vertices are chosen first, we get a different path.

c)
Time taken for IDS: 115.2136868131347 seconds
Total Memory Used for IDS: 110.046207 MB

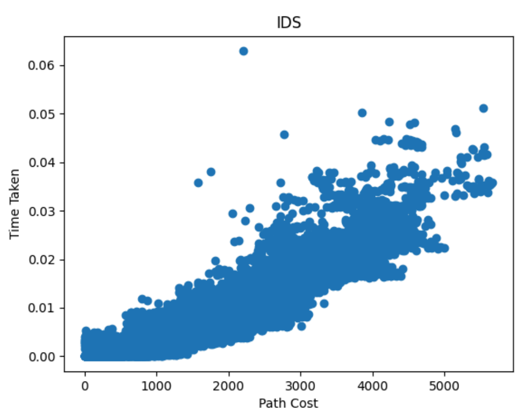
Time taken for Bidirectional BFS: 49.98461686773226 seconds
Total Memory Used for Bidirectional BFS: 33.345624 MB

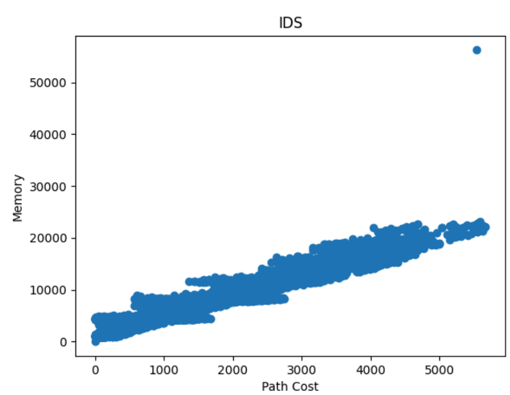
e) In the case of the informed algorithms, the Bidirectional A* Search algorithm, although implemented using the A* Search algorithm, might give different results, as this time we are running the algorithm from both the start and the end, so it is trying to find the shortest path according to the evaluation function from the end as well, which may not be the case for A* Search algorithm.

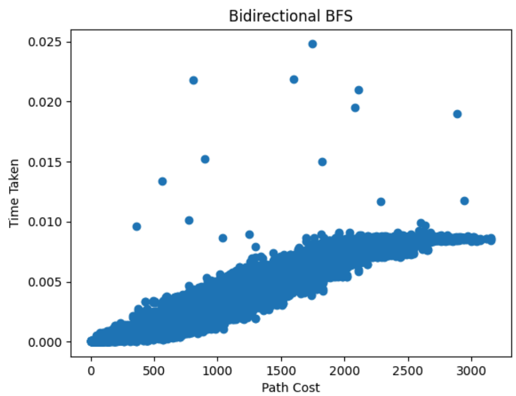
Time taken for A* Search: 73.26144528202713 seconds
Total Memory Used for A* Search: 107.112192 MB

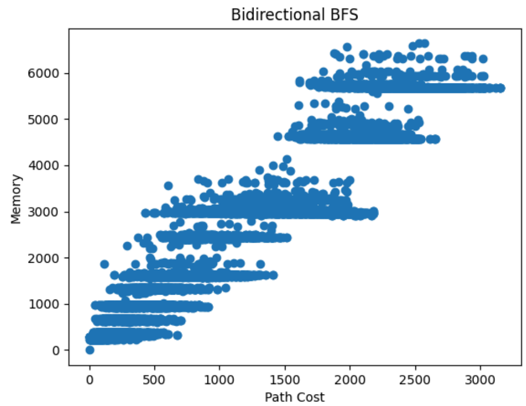
Time taken for Bidirectional A* Search: 70.74702090257779 seconds
Total Memory Used for Bidirectional A* Search: 97.845448 MB

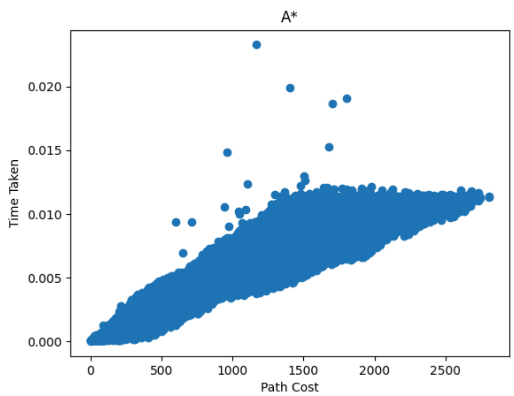
f)

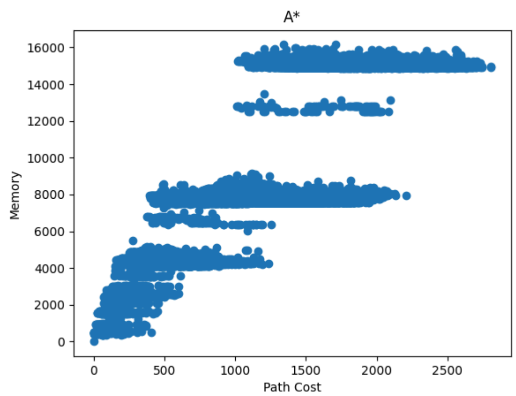


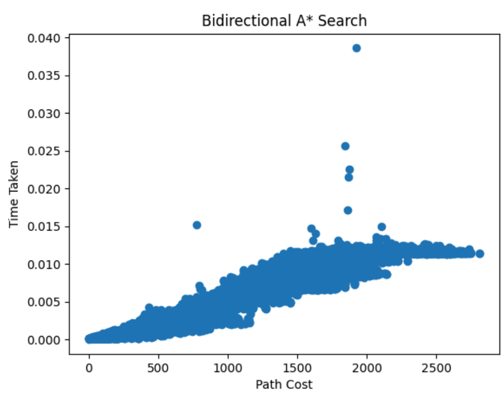


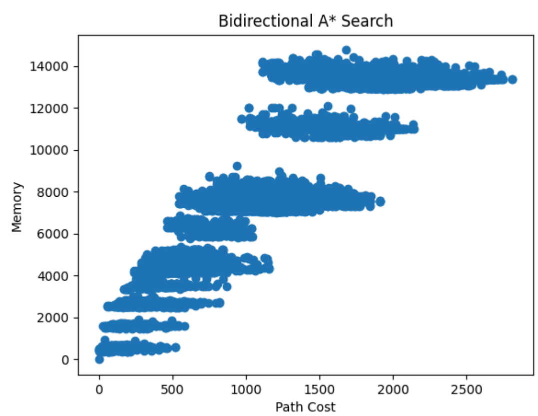












g) All vulnerable roads of the Graph are:

[(12, 57), (14, 53), (14, 99), (15, 35), (15, 46), (17, 45), (19, 100), (29, 42), (30, 42), (36, 38), (36, 114), (39, 40), (41, 70), (42, 113), (43, 113), (37, 44), (47, 48), (47, 49), (0, 49), (50, 51), (53, 54), (53, 95), (55, 56), (69, 124), (72, 73), (75, 106), (84, 114), (87, 88), (89, 90), (95, 96), (106, 107), (106, 111), (108, 109), (108, 111), (108, 112), (110, 111)]