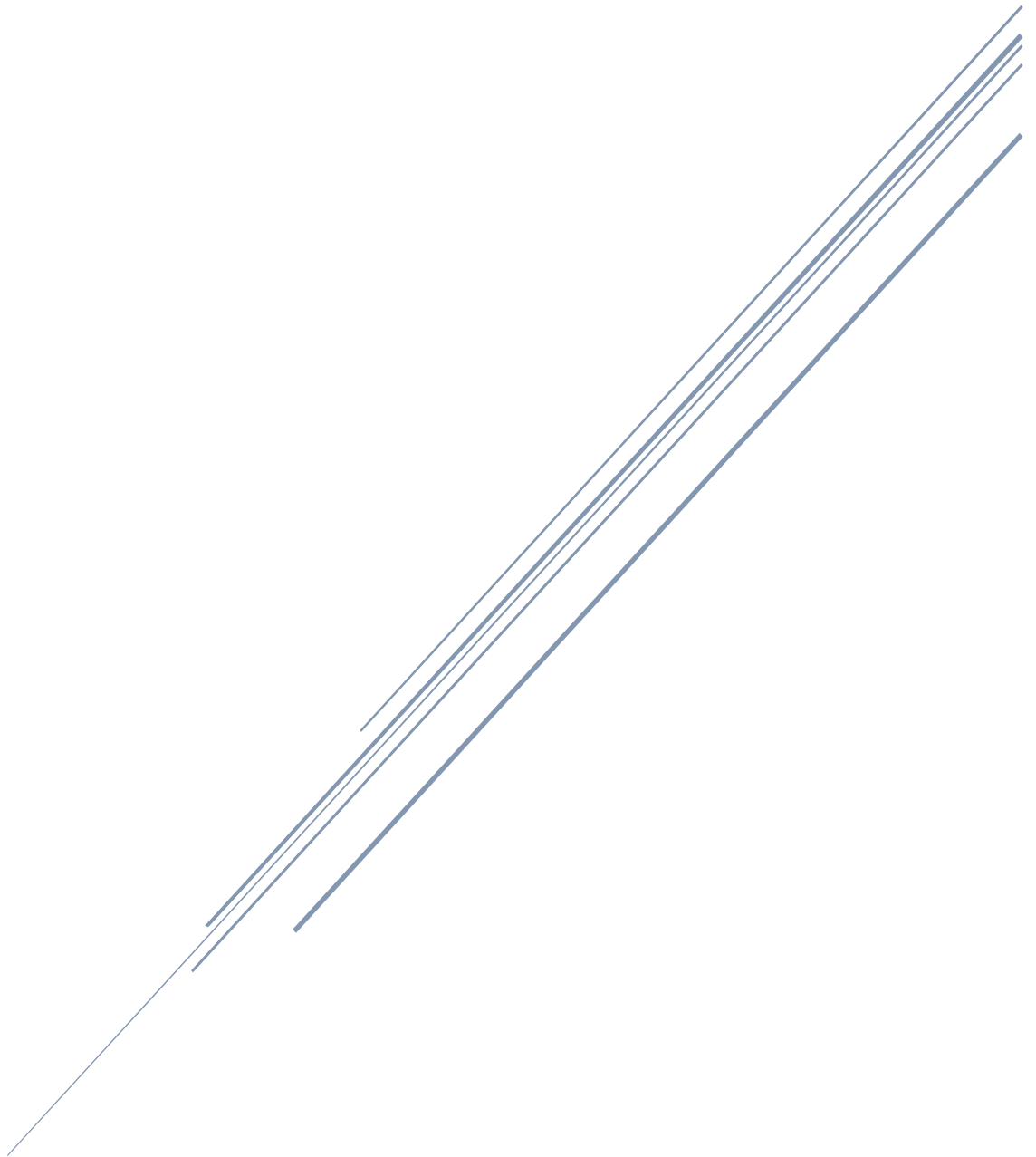


AI ASSIGNMENT 3 – UCS & IDS

Harsh Jha – 2311AI14

Kumar Satyam – 2311AI16

Supriya Maji – 2311AI20



Observation –

Goal Grid Configuration-

[1, 2, 3]

[4, 5, 6]

[7, 8, 0]

Note: Accumulated cost for IDS means all nodes searched at each depth iteration.

1. First Test-

Initial Grid Configuration-

[2, 5, 3]

[4, 8, 1]

[7, 0, 6]

Number of nodes searched and time -

BFS - 2480, Time - 0.06

DFS - 48595, Time - 1.63

UCS – 1178, Time - 0.04

IDS (Accumulated) – 2018, Time – 0.58

IDS (at depth where solution found) - 458

2. Second Test-

Initial Grid Configuration-

[5, 3, 8]

[0, 2, 7]

[6, 4, 1]

Number of nodes searched and time -

BFS - 113378, Time - 3.63

DFS - 162598, Time - 4.30

UCS – 112916, Time – 5.03

IDS (Accumulated) – 360804, Time – 10.14

IDS (at depth where solution found) - 38030

3. Third Test-

Initial Grid Configuration-

[8, 4, 6]

[3, 2, 5]

[7, 1, 0]

Note – For this initial grid config, no solution is found

Number of nodes searched and time -

BFS - 181441, Time - 5.66

DFS - 181441, Time - 4.66

UCS – 181440, Time – 8.87

IDS (Accumulated)– 1035594, Time – 39.45

4. Fourth Test-

Initial Grid Configuration-

[6, 7, 2]

[0, 5, 3]

[8, 4, 1]

Number of nodes searched and time -

BFS - 158973, Time - 5.03

DFS - 63095, Time - 1.87

UCS – 149949, Time – 6.58

IDS (Accumulated) – 411537, Time - 12.75

IDS (at depth where solution found) - 49952

Code Logic –

For both UCS and IDS, we are maintaining an explored list, which helps us to avoid loops and helps for duplicate checks (for IDS, we are generating a new explored list for each iteration).

Data Structure Used –

UCS – Stack

IDS – Priority Queue, (priority is based on level cost, i.e., cost from initial state till that node).

Unifor Cost Search (UCS)-

Main thinking and difference between UCS and BFS are that here we are using priority queue instead of queue (used in BFS). We are adding all nodes into priority queue based on cumulative cost of that node (i.e., number of steps taken from initial state to current node) and we are picking the least cost node from PQ to expand next.

We are repeating this until a solution is found or all nodes are visited.

Time Complexity – $O(b^{(c^*/\epsilon)})$

Space Complexity - $O(b^{(c^*/\epsilon)})$

Where b = branching factor

c^* = cost of the optimal path

ϵ = cost of action preformed

Iterative Deepening Search (IDS)-

One big issue with UCS/BFS is that space complexity is exponential and after a certain point we can say exponential space complexity is much worse than exponential space complexity.

DFS is better with space but it can go into infinite loop if we keep going in a depth and never return, to stop this we are keeping a list of explore states.

We are Depth Limited Search (DLS) (DFS but stop after a certain depth), till a solution is found or all nodes are visited and there is no further depth to go down.

IDS -

Time Complexity – $O(b^d)$

Space Complexity – $O(bd)$

DLS –

Time Complexity – $O(b^l)$

Space Complexity – $O(bl)$

Where b = branching factor

d = depth

l = depth limit for IDS where $l < d$

Based on above observations (test case 2), we can say BFS and UCS are very similar in terms of nodes searched.

BFS – **113378** and UCS – **112916**

But for IDS and DFS, we can observe a much-reduced number of nodes touched for the last depth where solution is found, which helps us in reducing space complexity, but overall number of nodes touched in IDS is much higher using which we can say Time Complexity higher for IDS, since we are touching many of the same nodes at each depth iteration.

DFS – **162598**, IDS (at depth where solution found) - **38030**

IDS (**Accumulated**) – **360804**