Name: Priyank Devpura **UIN:225002461**

Programming Assignment I

User Manual:

The programming is done using python

Command to run the code:

```
./eight.py --search <search algo> --level <level> --heu <h1,h2 for informed searches>
```

where search: bfs, dfs, ids, a-star, greedy, ida-star

level: easy, medium, hard

heu: h1, h2 required for informed searches(a-star, greedy, ida-star)

For running uninformed searches for eg. Bfs with level easy

bfs: Cmd: ./eight.py --search bfs --level easy

For running informed searches for eg. Greedy with level easy and heuristics h1

Greedy: ./eight.py --search greedy --level easy --heu h1

A look up table is made for inputs corresponding to easy, medium and hard

Easy	[1,3,4],[8,6,2],[7,0,5]
Medium	[2,8,1],[0,4,3],[7,6,5]
Hard	[5,6,7],[4,0,8],[3,2,1]

h1: heuristic which specifies number of inputs out of place

h2: heuristic which specifies the manhattan distance from state to goal

Results

Node are inserted in this order for all search algorithms: Up, left, down, right

Search Algo	**starting State**	**Maximum length of node list**	**Nodes visited**	**Path**
bfs	easy	36	43	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']
	medium	230	331	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']
	hard	73555	181357	['UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP']
dfs	easy	897	1147	Path is too long. Depth is 1139
	medium	58560	82482	Path is too long. Depth is 79013
	hard	5396	6919	Path is too long. Depth is 6872
ids	easy	8	169	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']
	medium	11	1225	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']
	hard	29	371314	['RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT']

Search	**Level**	**Max. nodes in a list**	**Nodes visited**	**Path**
greedy-h1	easy	11	10	['UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT']
	medium	185	280	['RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT']

	hard	654	982	['UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'UP', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT']
greedy-h2	easy	7	6	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']
	medium	19	18	['RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']
	hard	135	141	['UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'UP', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'UP', 'LEFT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'UP', 'LEFT', 'DOWN', 'RIGHT', 'UP', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN', 'RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT', 'UP']

Search Algo	**level**	**Maximum length of node list**	**Nodes visited**	**Path**
a-star-h1	easy	8	7	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']
	medium	23	24	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']
	hard	-	-	Long runtime
a-star-h2	easy	7	6	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']
	medium	17	17	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']
	hard	802	941	['RIGHT', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'DOWN', 'RIGHT', 'RIGHT', 'UP', 'UP', 'LEFT', 'LEFT', 'DOWN', 'RIGHT']

Search Algo	**level**	**Maximum length of node list**	**Nodes visited**	**Path**	**Depth of recursion**
ida-star-h1	easy	6	19	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']	5
	medium	19	116	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']	9
	hard	-	-	Long runtime	
ida-star-h2	easy	4	21	['UP', 'RIGHT', 'UP', 'LEFT', 'DOWN']	5
	medium	9	55	['UP', 'RIGHT', 'RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT', 'DOWN']	9
	hard	806	8808	['RIGHT', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT', 'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'UP', 'RIGHT', 'RIGHT',	30

				'DOWN', 'DOWN', 'LEFT', 'LEFT', 'UP', 'RIGHT']	

Analysis

A-star with the h1 heuristics does not produce result for the hard level as heuristics is not that accurate towards achieving the required goal. It causes A-star to traverse the wrong path leading to higher time complexity. Sometimes with increase in depth (g) heuristics value decreases and at lower depth heuristics value increases keeping f value around same, thus leading to insertion of a lot of nodes in the queue.

Similarly for ida-star with the h1 heuristics does not produce result for the hard level as heuristics is not that accurate in achieving the desired goal. As the f-cut off value from 25 for the contour increases lot of nodes are inserted in the queue and thus taking a lot of time to process each one. The h1 heuristics is not very close to the actual value than h2 where we get the result quickly.

- From table we can make out that h2 heuristics is better than h1 heuristics for informed search as it is closer to the actual value.
- Time Complexity is calculated based on the total number of nodes inserted in the fringe(queue or stack) as to process each node we have to search the visited list. Lower the number of nodes will lead to lower time complexity. It is assumed that heuristics calculation is very simple and does not take time.

  Easy Level Time Complexity Order: a-star<greedy<ida-star<bfs<ids<dfs

  Medium Level Time Complexity Order: ida-star<a-star<greedy<bfs<ids<dfs

  Hard Level Time Complexity Order: greedy<a-star<ida-star<ids<dfs<bfs

  - From this, we can say that informed searches takes lesser time than uninformed searches which is expected as heuristics guide it towards the goal in a better way. So better the heuristics, more quickly we will get the result with informed searches.

- Space Complexity is calculated by calculating the maximum number of elements in the queue/stack.

  Easy Level Space Complexity: ida-star<a-star<ids<greedy<bfs<dfs

  Medium Level Space Complexity: ida-star<ids<a-star<greedy<bfs<dfs

  Hard Level Space Complexity: ids<greedy<ida-star<a-star<dfs<bfs

  - The dfs takes a lot space when we go at a deeper depth. This really happened in case of dfs hard and medium level.
  - Bfs in case of hard level takes the maximum space as the solution is achieved at level 30 and since bfs as exponential space complexity which increases as we increase levels thus taking a lot of space.
  - The ids is mixed of dfs and bfs and thus takes lesser space as we store nodes of a particular depth.

  - The informed searches such as A-star, greedy have lower space complexity than bfs and dfs we keep going in the correct path with lesser insertion of nodes.
- Total number of nodes visited is more in ids in comparison to bfs as we are visiting node each time when the new iteration begins.
- Similarly ida-star has more number of visited nodes in comparison to a-star.

Strengths/Weakness

Search method	**Strength**	**Weakness**
bfs	Optimal and Complete Solution	Exponential space O(b^d) and time complexity O(b^d)
dfs	Low space complexity as it stores nodes which are there in the current path. It is good when goal are not very deep.	Non-optimal and not complete solution. A very large tree make dfs not practical
ids	Mix of dfs and bfs. Optimal and Complete Solution. Lower space complexity in comparison with bfs. This is useful for the cases when the size of the tree is unknown.	Exponential time complexity and visiting search nodes again and again for different iterations
greedy	Lower search time compared to uninformed search if heuristics is good.	Not optimal and not complete and may need to backtrack due to bad heuristics and can give a non-optimal result
a-star	Complete and optimal solution. Better space and time complexity if the heuristics are good.	Bad heuristics can result in the search to become exponential as we saw in case of h1 for hard case.
Ida-star	Complete and optimal solution. Better space and time complexity if heuristics are good.	States with widely varying heuristics causes issues in IDAstar as it starts including only one new node per contour .This blows up the space complexity to O(N^2)