Uninformed and Informed Search

Artificial Intelligence(CS5100) - HW2

Nephi Calvin Bandela

1. Search Techniques Implemented:

A. Uninformed Search:

i. Depth First Search:

a. Algorithm:

```
1. Initialize Stack
2. Push Start Node into Stack
3. Mark Start Node as Visited
 loop
4. Get Top
5. Get Children of Top
6. If Child Node Unvisited
7. Mark Unvisited child as visited
8. Push Unvisited child onto Stack
9. If no child nodes - Retract Move
 end loop
```

b. Complexity: b-branching factor, m: Max depth from start, d: Goal depth or solution depth

Time Complexity	Space Complexity
O(b^m) – May visit all nodes- b+b^2+…….+b^m	O(bm) which is m*(b-1) -Entire Space Storage

c. Environment: Partially Observable

B. Informed Search:

ii. Greedy Best First Search

a. Algorithm:

```
1. Retrieve all Node Indices of Targets
2. Initialize Visited, HashMap to store heuristics and their corresponding Node Indices
3. The heuristics- Manhattan Distance are sorted in ascending order.
4. The HashMap consists of successors of current Node arranged in increasing orders
5. Return the Node with the best heuristic as the target node
```

b. Complexity: b-branching factor, m: Max depth from start, d: Goal depth or solution depth. h(n)- Heuristic Function

Time Complexity	Space Complexity
O(b^m) – May visit all nodes- b+b^2+…….+b^m but due to the heuristic function h(n) -Manhattan Distance – It might be almost O(b^d)	O(bm) which is m*(b-1) - Entire Space Storage

c. Environment: Fully Observable

iii. AStar Search:

a. Algorithm:

Note: Algorithm implementation taken from the PacMan framework's game internals.

```
1. Construct the graph with new node type N(contains g,h) by passing the current maze's graph
2. Initialize open
3. Initialize closed
4. Place start node into open
5. loop - While open not empty
6. poll the open list(the node with least f=g+h)
7. find popped nodes children
8. for each child
9. if successor is the target then break
10. child's gvalue=popped node's g+distance between the child and popped node
11. child's fvalue=gvalue+h(heuristic measure)
12. If the node same as the child is in open list with lower f-skip the node
13. do as above for closed list
14. else add the child to open list
15. return the path to reach target
```

b. Complexity: b-branching factor, m: Max depth from start, d: Goal depth or solution depth. h(n)- Heuristic Function

Time Complexity	Space Complexity
O(b^d) – May visit all nodes- b+b^2+…….+b^d where d is the solution depth. This is better than the complexity of Greedy BFS	O(bm) which is m*(b-1) -Entire Space Storage

c. Environment: Fully Observable

2. References:

a. http://www.codeproject.com/Articles/32212/Introduction-to-Graph-with-Breadth-First-Search-BF

b. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach.

c. PacMan Internal framework.