* Shiva Saran
* BE-B 20

Artifical Inteligence & Robotics
Assingnment 1

Q1. What is AI? State & explain various applications areas of AI.

→ • AI is the study of How to make computers do things which at the moment people can do better.

• AI is the Intelligence of machines and brach of computer science that aims to create it.

Application areas:

① Game playing
Programing computers to play games against human opnents.
Eg. Chess

② Expert System
Programming computers to make decision in real-life situations.
Eg. Flight tracking, clinical systems.

③ Natural languages
Programming computers to ~~make decision~~ understand natural human language.
Eg. Siri, Alexa, Google now, Voice to text

④ Neural Network
System that simulate intelligence by attempting to reproduce the types of physical connetions. that occur in animal brains.
Eg. ~~Rage~~ Recognition of face, handwriting, text, images

(5) Robotics:

It deals with the design, construction, operation & application of robots, as well as computer systems for their control, sensory feedback & information processing.

Q2. Explain Depth Bounded DFS & Depth first Iterative deepening methods with an example.

→ * Depth Bounded DFS / Depth limited Search

- Selects some limit in depth to explore the problem using DFS.
- Selecting the depth solves infinite path problems.
- DFS can be viewed as a special case of depth-limited search with l to infinity.
- To overcome the infinite length drawback.
- Its Not-complete - since the solution may not be found in all cases. It is complete when depth limit is greater than that of solution depth.
- Not optimal.

Time complexity: $O(b^l)$
Space complexity: $O(b^l)$

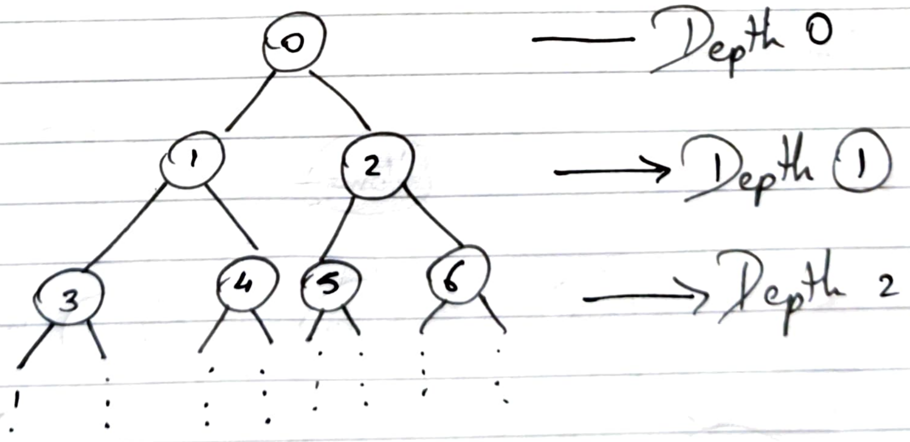
* Iterative Depth first Search

- Its a search strategy resulting when you combine BFS & DFS, thus ~~com~~ combing the advantages of each strategy, taking the completeness and optimality of BFS & the modest memory requirements of DFS.
- IDS works by looking for the search depth d, thus stating with depth limit 0 & make a BFS.

- If the search failed, it increased the depth limit by 1 & try a BFS again & so-on.
- Like BFS, IDS is complete when branching factor b is finite.
- IDS is optimal.
- Time complexity of IDS is $O(b^d)$
  Space complexity $O(b*d)$.

Question 3

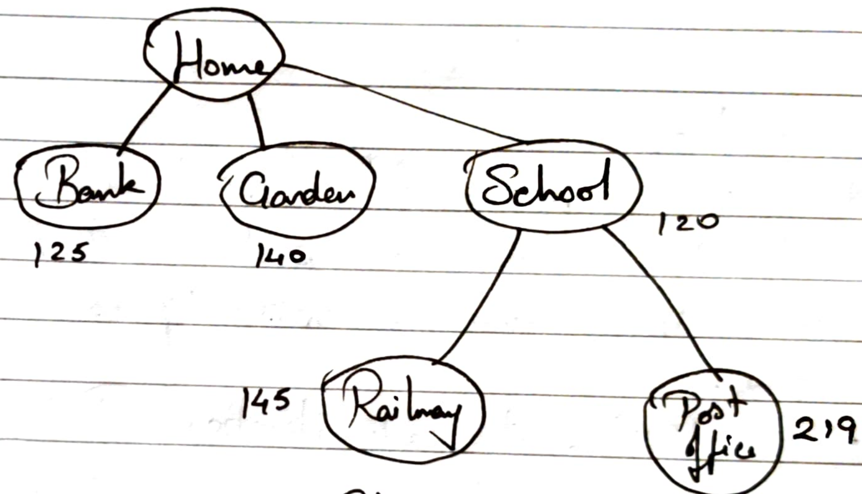
* Iterative Deepening A^* (IDA^*)
  - IDA^* resolves the problem of A^* where the memory issue is overcome & optimality is maintained at the same time.
  - In IDA^*, at each iteration DFS is applied. A track is maintained of the costs i.e. $f(n) = g(n) + h(n)$ of each & every node that is generated.
  - When ever a node is generated where cost is more than the threshold of that iteration, the path is discarded.



* Recursive Best Search

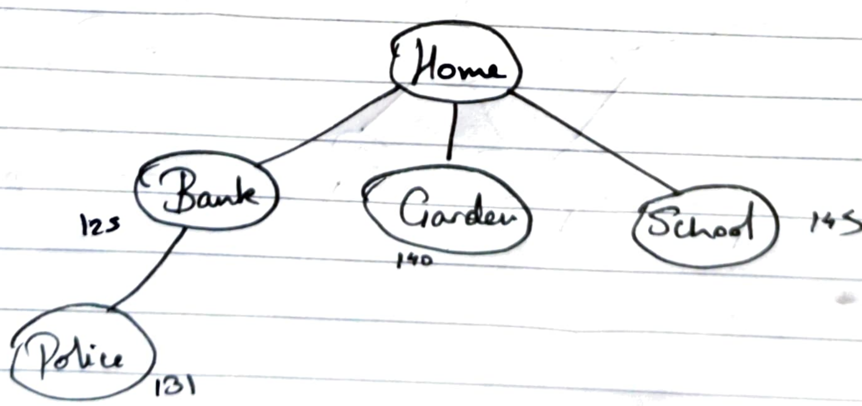
- The basic idea here is to remember the best path or the best alternative (i.e. lowest $f(n)$) node. Backtrack if the best first gets very expensive i.e. when the cost exceeds that of previously expanded node.

Eg.



Stop expansion & backup. Since $f(n)$ of Railway station & post office is more than $f(n)$ school.

Hence wee get



Stop expansion & backup

Question 4

* Tabu Search

- The idea of Tabu search is to continue searching instead of terminating when there are no better choices are available.
- Tabu Search is metaheuristic local search method used for mathematical optimization.
- It does not terminate on reaching maximum, instead in continues to search until some criterion is met.
- Prohibition (tabu) are introduced to discourage the search from coming back to previously visited solution ie. shouldn't consider the solution again.
- When a tabu move has a sufficiently attractive evaluation where it could result in a solution better than any visited so far, then its tabu classification may be overidden. A condition that allows such an override to occur is called an Aspiration Criterion.
- Tabu list is used for recently visited states & and are temporarily excluded from being visited again.

* Advantages:

① It allows to exit from sub-optimal regions by making non-improving solution to be accepted.
② Use of tabu list improves efficiency.

* Disadvantage:

Can't find global optimum in some cases.

Question 5
A* Search

- A* is a combination of uniform cost search & greedy search.
- Heuristic function of A* is

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

where $g(n)$ = actual path cost from start state to node n.
$h(n)$ = Heuristic /estimated path cost from node n to goal state.

- A* Search is complete & guarantees a solution.
- It is optimal if $h(n) <=$ cost to goal.
- Time complexity depends on heuristic function.
- Not suitable for large scale problems.

* 8 Puzzle Problem using A* Search

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

① Initial State

2	8	3
1	6	4
7	–	5

Final State

1	2	3
8	–	4
7	6	5

$g=0$ $h=6$ $f=6$

2	8	3
1	6	4
7	–	5

Left

2	8	3
1	6	4
–	7	5

$g=1$ $h=6$ $f=7$

Up

2	8	3
1	–	4
7	6	5

$g=1$ $h=3$ $f=4$

Right

2	8	3
1	6	4
7	5	–

$f=8$ $g=1$ $h=7$

Up

2	–	3
1	8	4
7	6	5

$g=2$ $h=4$ $f=6$

Left

2	8	3
–	1	4
7	6	5

$g=2$ $h=4$ $f(n)=6$

–	2	3
1	8	4
7	6	5

$g=3$ $h=3$ $f=6$

2	3	–
1	8	4
7	6	5

$g=3$ $h=5$ $f=8$

–	8	3
2	1	4
7	6	5

$g=3$ $h=4$ $f=7$

2	8	3
7	1	4
–	6	5

$g=3$ $h=5$ $f=8$

–	2	3
1	8	4
7	6	5

2	–	3
1	8	4
7	6	5

$g = 4 \quad h = 4 \quad f = 8$

1	2	3
–	8	4
7	6	5

$g = 4 \quad h = 2 \quad f = 6$

$\downarrow$

1	2	3
8	–	4
7	6	5

$g = 5 \quad h = 0 \quad f = 5$

Goal State