Artificial Intelligence

Module 2: Informed Search- Generate and Test, Best First Search, Heuristics Search, A*, Problem reduction, AO*, Constraint Satisfaction problems, Hill climbing, Simulated annealing.
Adversarial Search: Min-max search, Alpha beta cut-offs.

INFORMED (HEURISTIC) SEARCH STRATEGIES

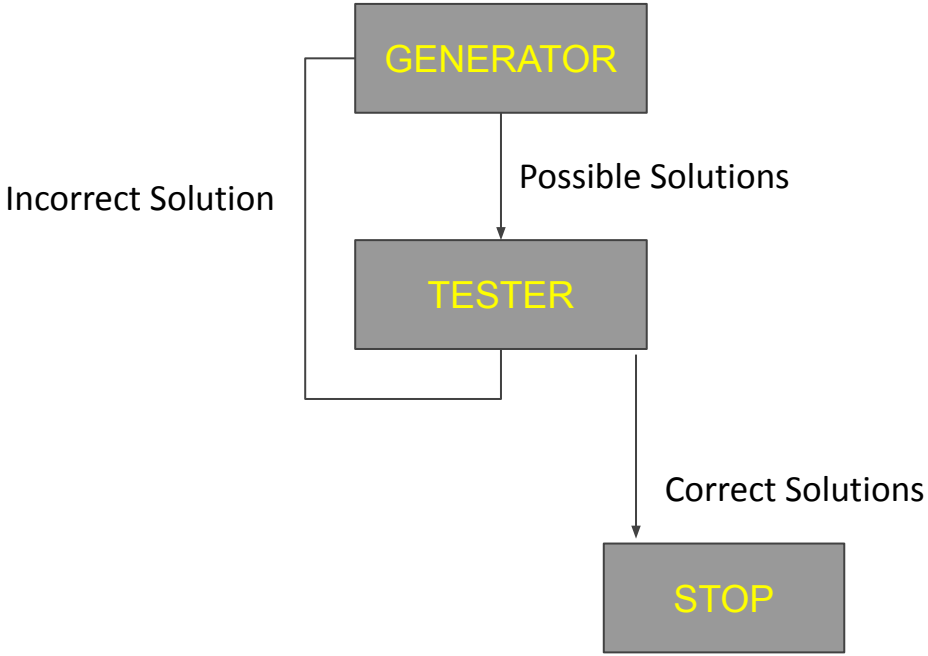
- The **informed search strategy** shows one that uses **problem-specific knowledge** beyond the definition of the **problem itself** can **find solutions more efficiently** than can an uninformed strategy.
- A **heuristic** is a way which might not always be guaranteed for best solutions but guaranteed to find a good solution in reasonable time.
- An example for informed search algorithm is Travelling Salesman Problem.

Generate and Test Algorithm

- Generate and Test strategy is the simplest approach.
- It uses the concept of **DFS with backtracking.**
- In this technique all the solutions are generated and tested for the best solution.
- It’s also known as **British Museum Search Algorithm** as it’s looking for an exhibit at random or finding an object in the British Museum by wandering randomly.

Algorithm

1. ***Generate*** *a possible solution. For example, generating a particular point in the problem space or generating a path for a start state.*
2. ***Test*** *to see if this is a actual solution by comparing the chosen point or the endpoint of the chosen path to the set of acceptable goal states*
3. *If a solution is found,* ***quit****. Otherwise go to Step 1*


GENERATOR
Possible Solutions
Incorrect Solution
TESTER
Correct Solutions
STOP

Properties of Good Generator

- ***Complete:*** *Good Generators need to be complete i.e. they should generate all the possible solutions and cover all the possible states. In this way, we can guaranty our algorithm to converge to the correct solution at some point in time.*
- ***Non Redundant:*** *Good Generators should not yield a duplicate solution at any point of time as it reduces the efficiency of algorithm thereby increasing the time of search and making the time complexity exponential. In fact, it is often said that if solutions appear several times in the depth-first search then it is better to modify the procedure to traverse a graph rather than a tree.*
- ***Informed:*** *Good Generators have the knowledge about the search space which they maintain in the form of an array of knowledge. This can be used to search how far the agent is from the goal, calculate the path cost and even find a way to reach the goal.*

Real Life Example

- *Dendral which infers the structure of organic compounds using NMR spectrogram also uses* ***plan-generate-test.***

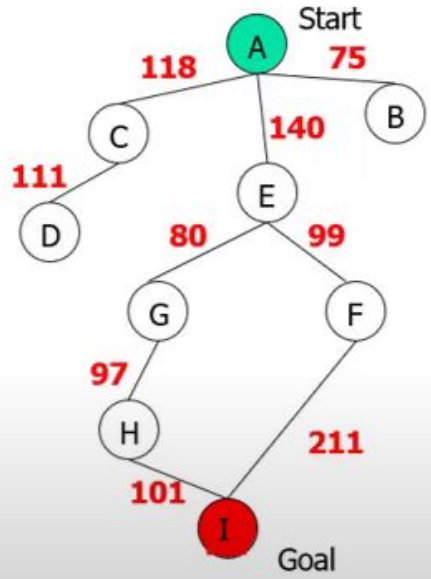
GREEDY BEST FIRST SEARCH

- The Best First Search algorithm always select the path which appears best at the moment.
- It is the combination of depth-first search and breadth-first search algorithms.
- Is uses the heuristic function and search.
- With the help of the best-first search, at each step, we can choose the most promising node.
- In the best first search algorithm, we expand the node which is closest to the goal node and the minimum cost is estimated by heuristic function.

GREEDY BEST FIRST SEARCH

- The evaluation function is f(n) = h(n)
- Where h(n) = estimated cost from node n to the goal.
- Greedy search ignores the cost of the path that has already been traversed to reach n
- Therefore, the solution given is not necessarily optimal.

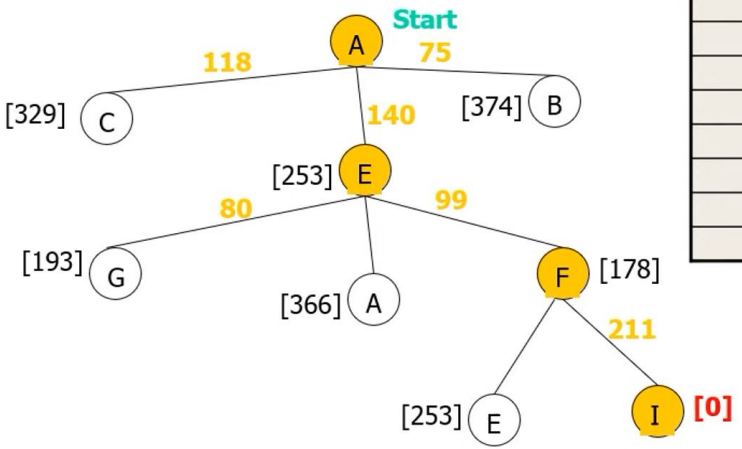
GREEDY BEST FIRST SEARCH



State	Heuristic: h(n)
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

***f(n) = h(n)* = straight-line distance heuristic**

GREEDY BEST FIRST SEARCH



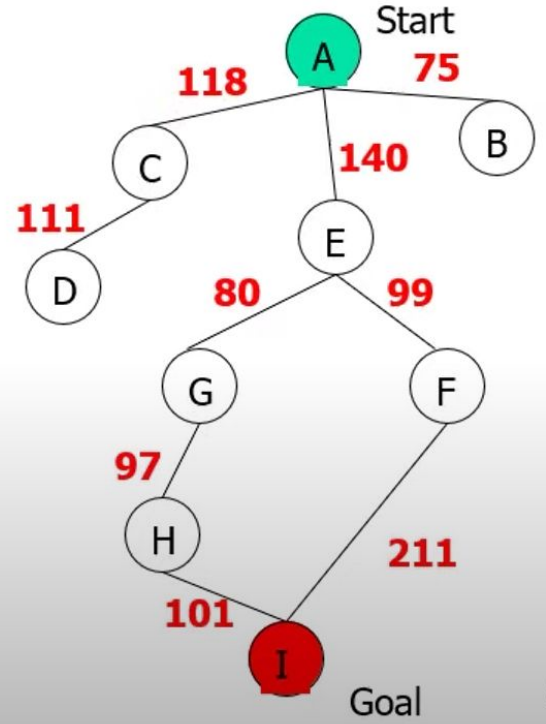
State	Heuristic: h(n)
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

Path cost(A-E-F-I) = 253 + 178 + 0 = 431

dist(A-E-F-I)= 140 +99+211=450

111

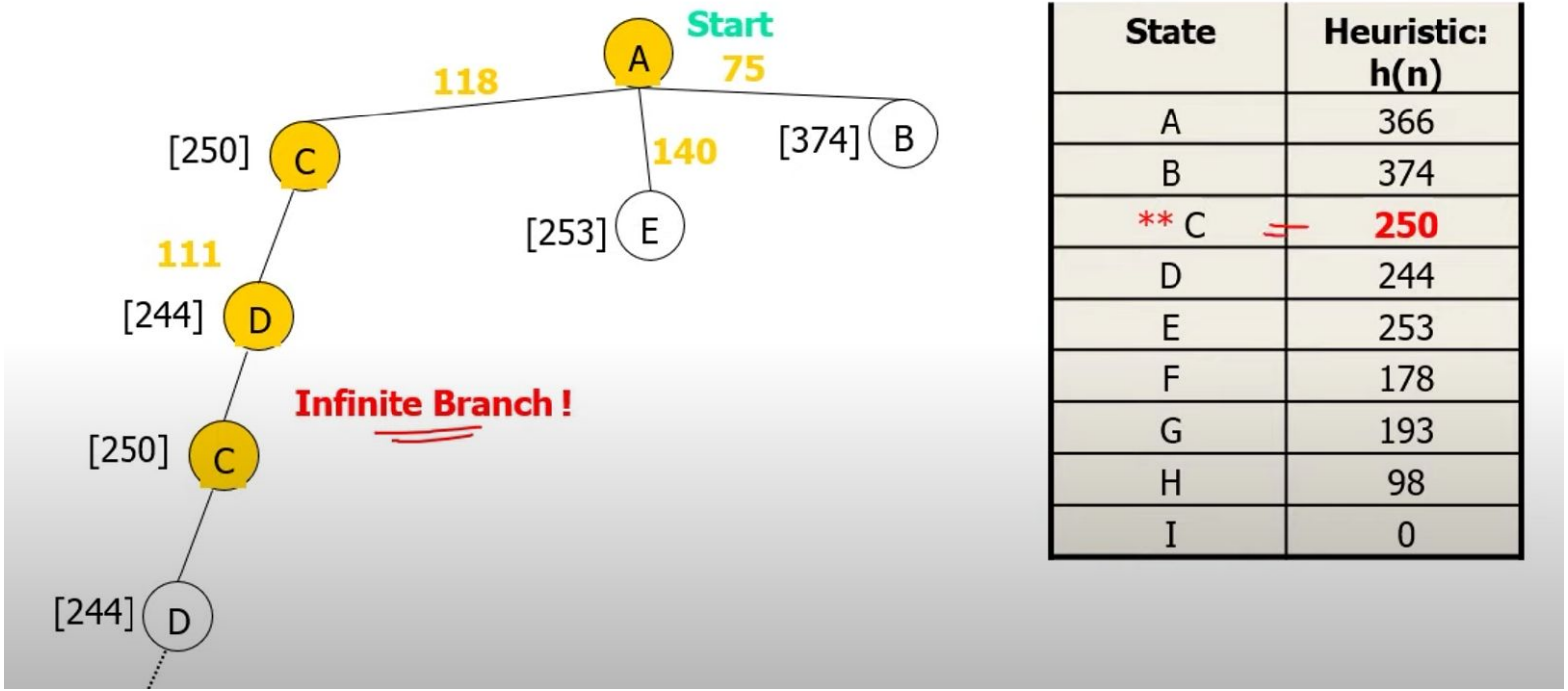
GREEDY BEST FIRST SEARCH



State	Heuristic: h(n)
A	366
B	374
** C	250
D	244
E	253
F	178
G	193
H	98
I	0

f(n) = h(n) = straight-line distance heuristic

GREEDY BEST FIRST SEARCH



State	Heuristic: h(n)
A	366
B	374
** C	250
D	244
E	253
F	178
G	193
H	98
I	0

GREEDY BEST FIRST SEARCH

- Greedy best-first search can start down an infinite path and never return to try other possibilities, it is incomplete.
- Because of its greediness the search makes choice that can lead to a dead end; then one backs up in the search tree to the deepest unexpanded node.
- Greedy best-first search resembles depth-first search in the way it prefers to follow a single path all the way to goal, but will back up when it hits a dead end.
- The quality of the heuristic function determines the practical usability of greedy search.

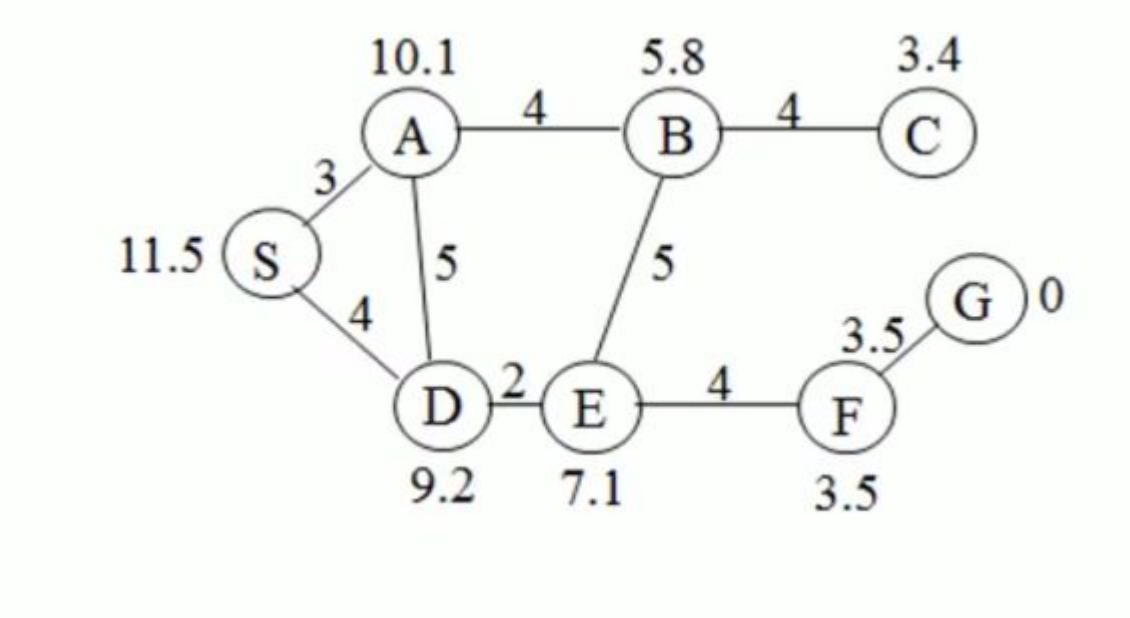
GREEDY BEST FIRST SEARCH

- Greedy search is **not optimal.**
- Greedy search is **incomplete** without systematic checking of repeated states.
- In the worst case, the **Time and space complexity** of Greedy Search are both **$O(b^m)$,**
- Where
  - **b is the branching factor and**
  - **m the maximum path length.**

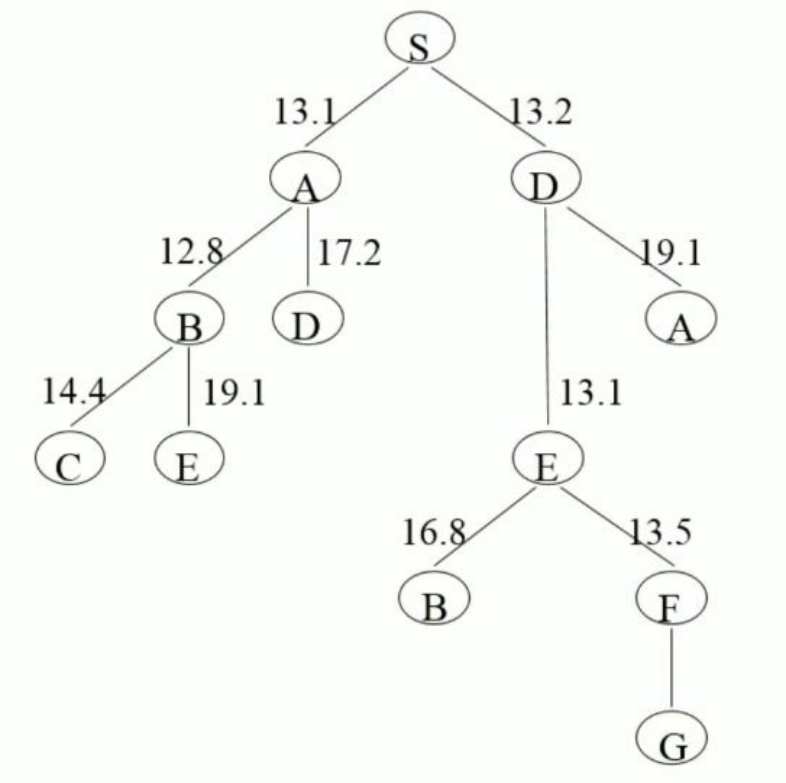
A* (star) Search Algorithm

- Algorithm A* (1968):
  - **f(n) = g(n) + h(n)**
- h(n) = cost of the cheapest path from node **n** to a **goal state**
- g(n) = cost of the cheapest path from the **initial state** to node **n**

A* (star) search Algorithm



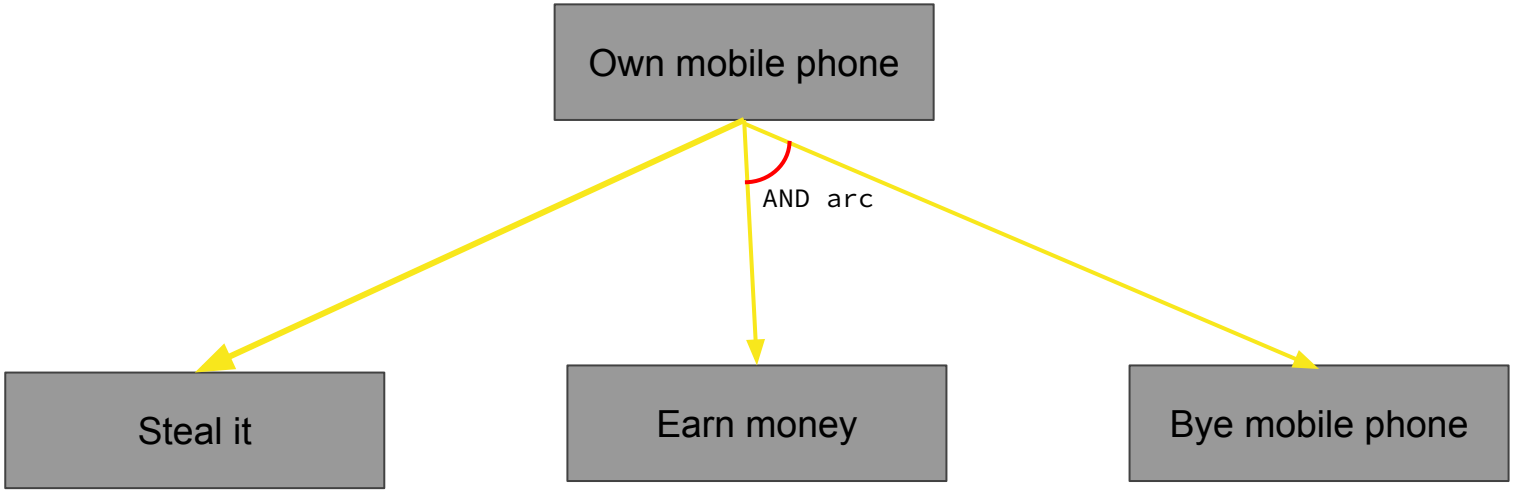
Cont.



AO* Algorithm

- It’s an informed search and works as best first search.
- AO* algorithm is based on problem decomposition(breakdown problem into small pieces).
- It represents AND-OR graph algorithm that is used to find more than one solution.
- The AO* algorithm is a knowledge-based search technique, meaning the start state and the goal state is already defined , and the best path is found using heuristics.
- The time complexity of the algorithm is significantly reduced due to the informed search technique.Compared to the A* algorithm , AO* algorithm is very efficient in searching the AND-OR trees very efficiently.

AND OR Graph


Own mobile phone
AND arc
Steal it
Earn money
Bye mobile phone

Working of AO* algorithm

The AO* algorithm works on the formula given below :

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

where,

- **g(n):** The actual cost of traversal from initial state to the current state.
- **h(n):** The estimated cost of traversal from the current state to the goal state.
- **f(n):** The actual cost of traversal from the initial state to the goal state.

Problem