AI Essentials

Search Algorithms

ir. Hennion Domien

Search Algorithms

Definition

- A search problem consists of:
  - **A State Space**: Set of all possible states where you can be.
  - **A Start State**: The state from where the search begins.
  - **A Goal Test**: A function that looks at the current state returns whether or not it is the goal state.
- The **Solution** to a search problem is a sequence of actions, called the **Plan** that transforms the start state to the goal state.
- The **Plan** is determined by a search algorithm

Uninformed Search

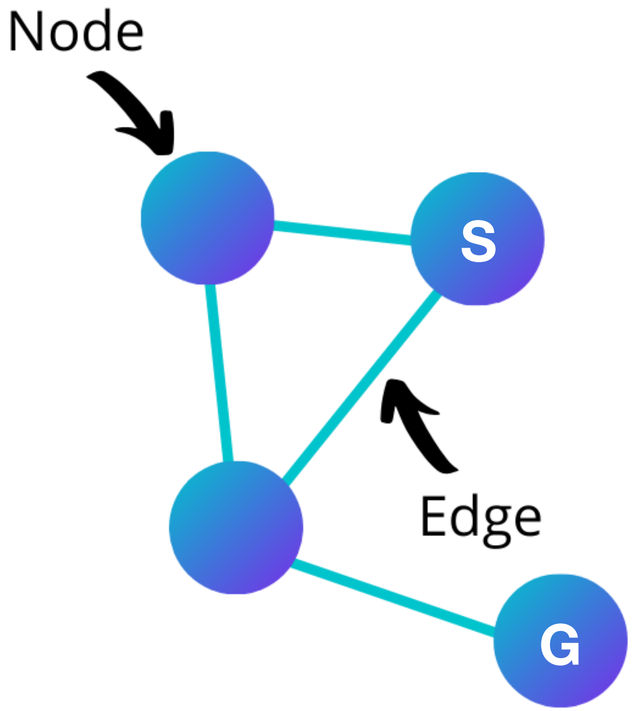
Search Algorithms

- No additional information on the goal node other than the one provided in the problem definition.
- The plans to reach the goal state from the start state differ only by the order and/or length of actions
- Blind search

Uninformed Search

Search Algorithms

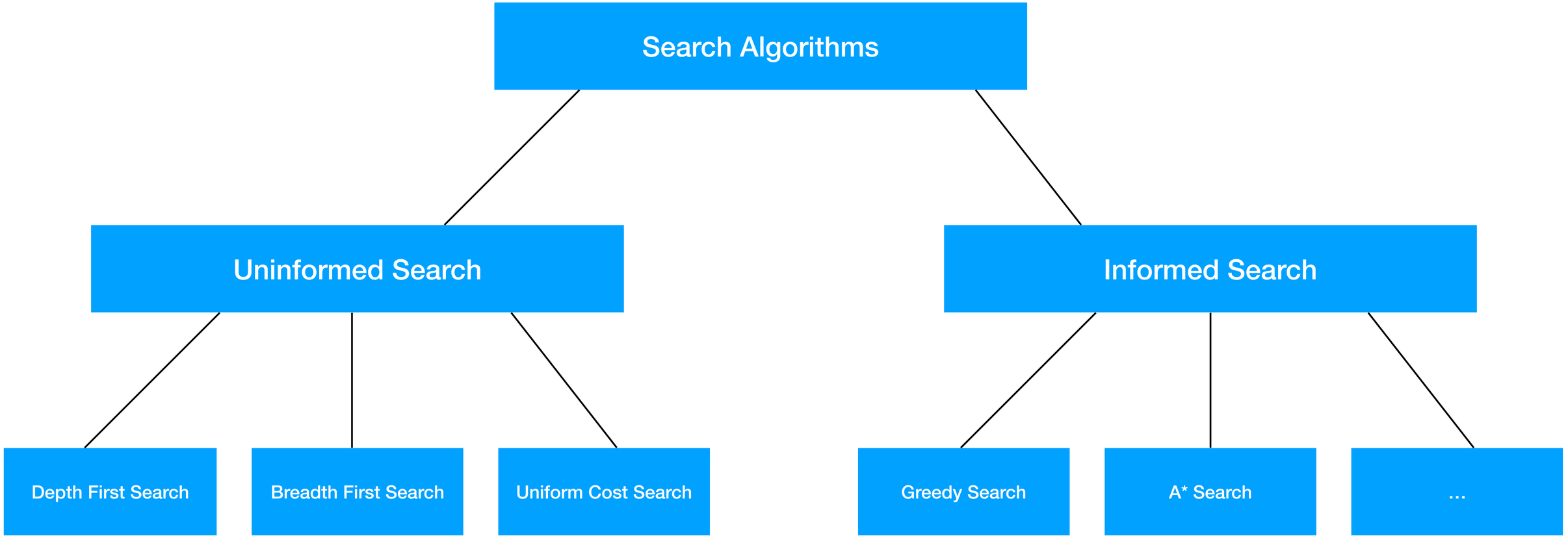
- These algorithms have:
  - A problem graph with **start node S** and **goal node G**
  - A **strategy** describing how to travers the graph
  - A **fringe**, the data structure representing the graph
  - A sequence from node S to G called the solution **plan**



Informed Search

Search Algorithms

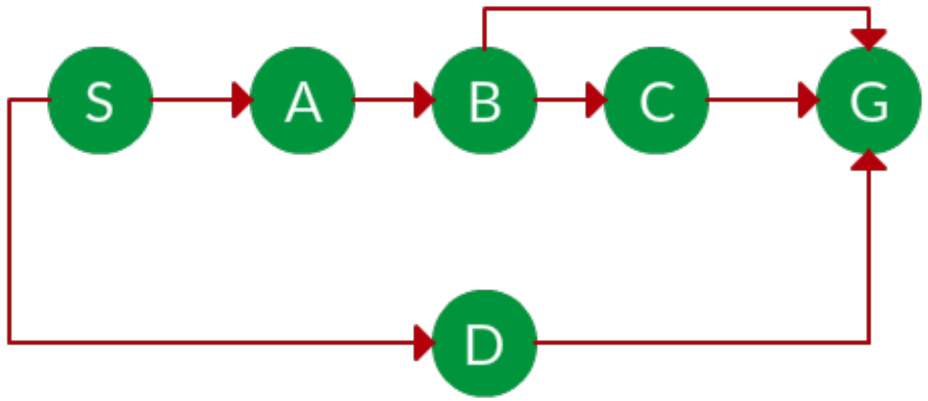
- More information
- Helps for more efficient searching
- The information is obtained by a **heuristic**
- A heuristic is a function that estimates how close a state is to the goal state.

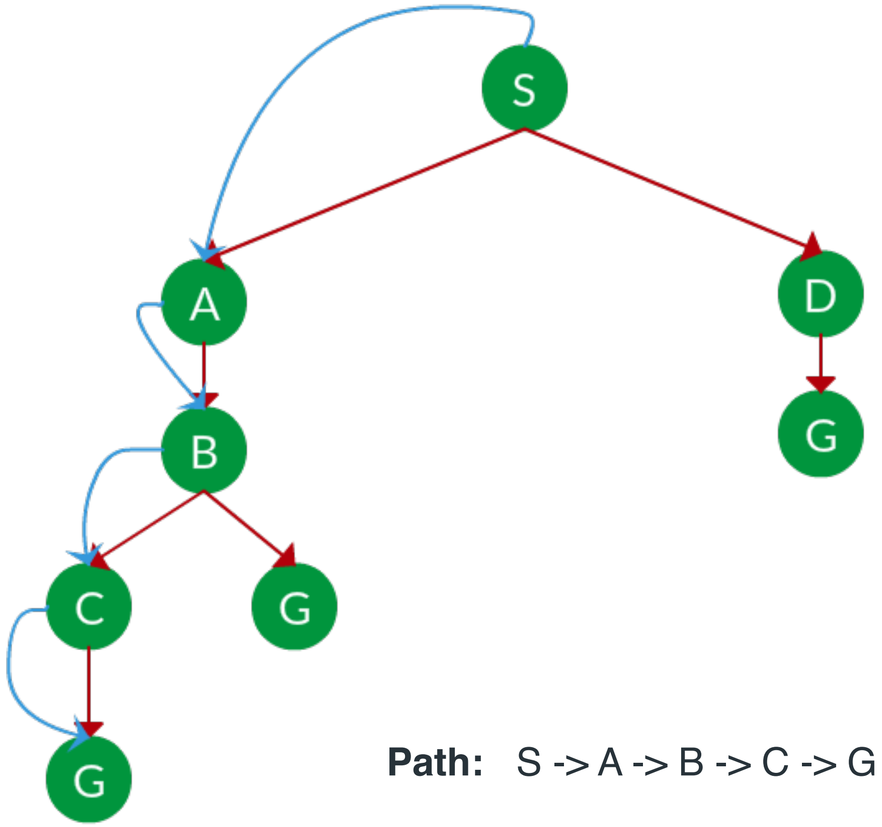

Search Algorithms
Uninformed Search
Informed Search
Depth First Search
Breadth First Search
Uniform Cost Search
Greedy Search
A* Search
...

Depth First Search

Uninformed Search

- Exploring as far as possible along each branch before backtracking



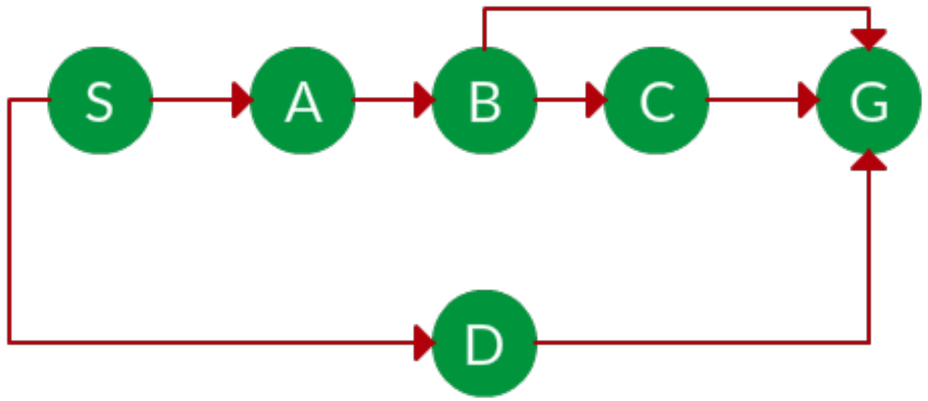


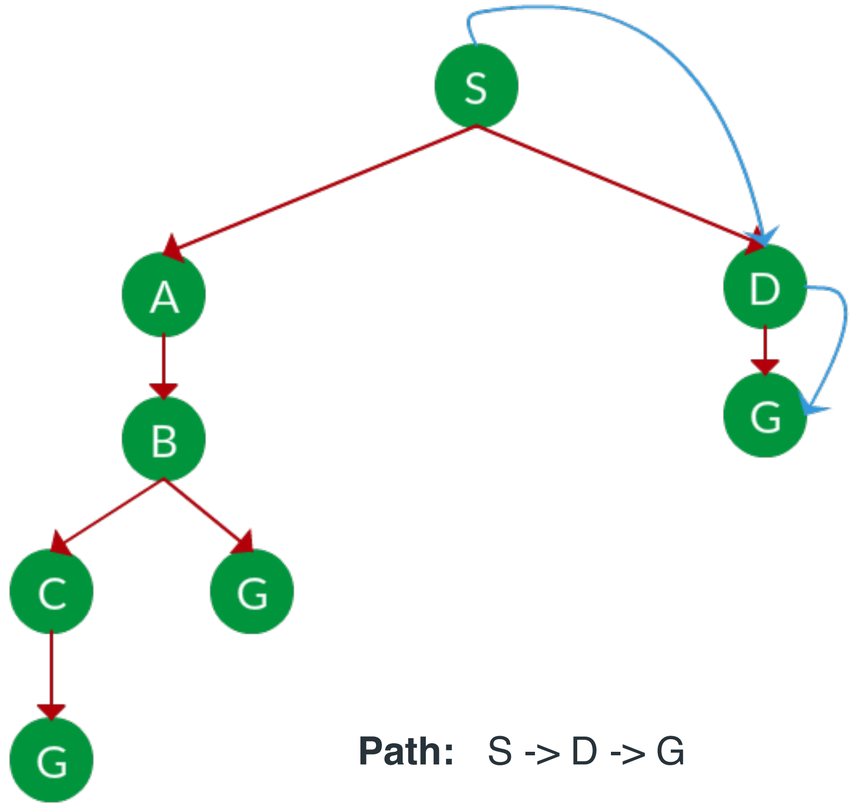
Path: S -> A -> B -> C -> G

Breadth Frist Search

Uninformed Search

- Exploring all of the neighbour nodes a the present depth prior to moving on to the nodes at the next depth level



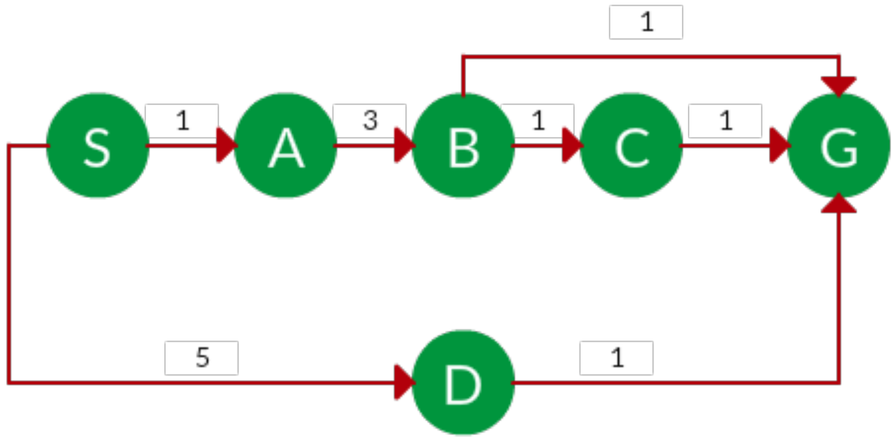


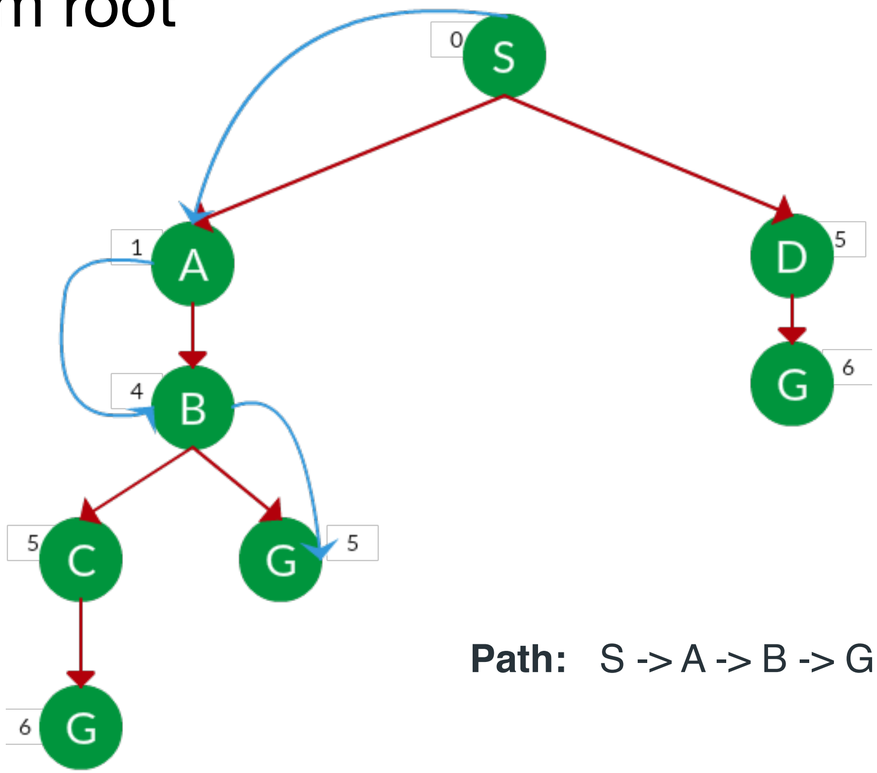
Path: S -> D -> G

Uniform Cost Search

Uninformed Search

- The goal is to find the path where the total costs is the least
- cost(node) = cumulative cost of all nodes from root

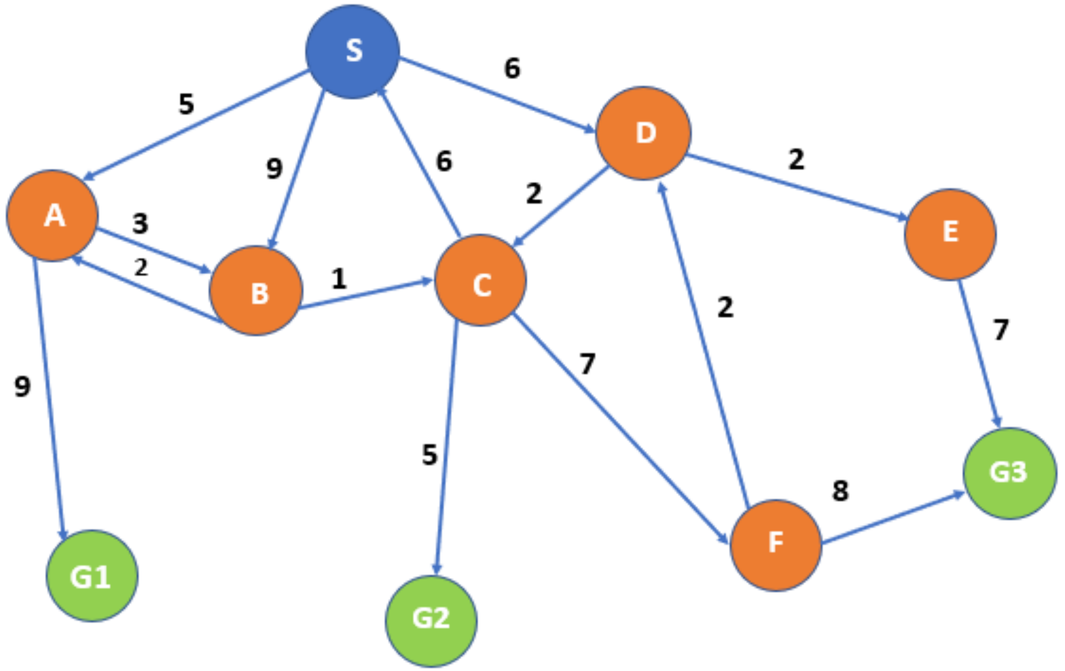


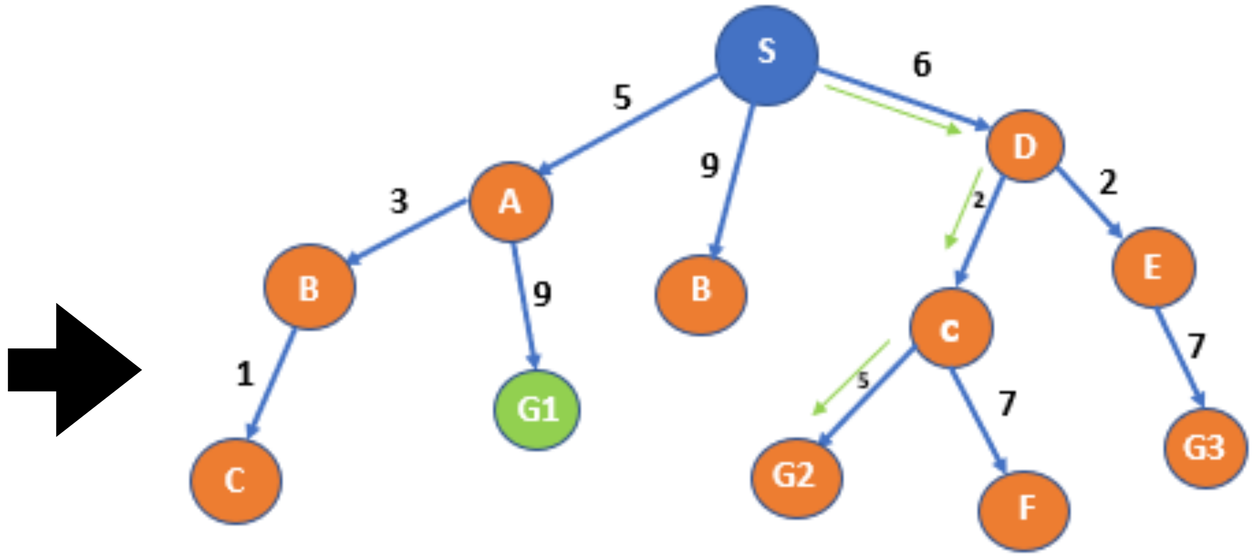


Path: S -> A -> B -> G

Uniform Cost Search

Uninformed Search



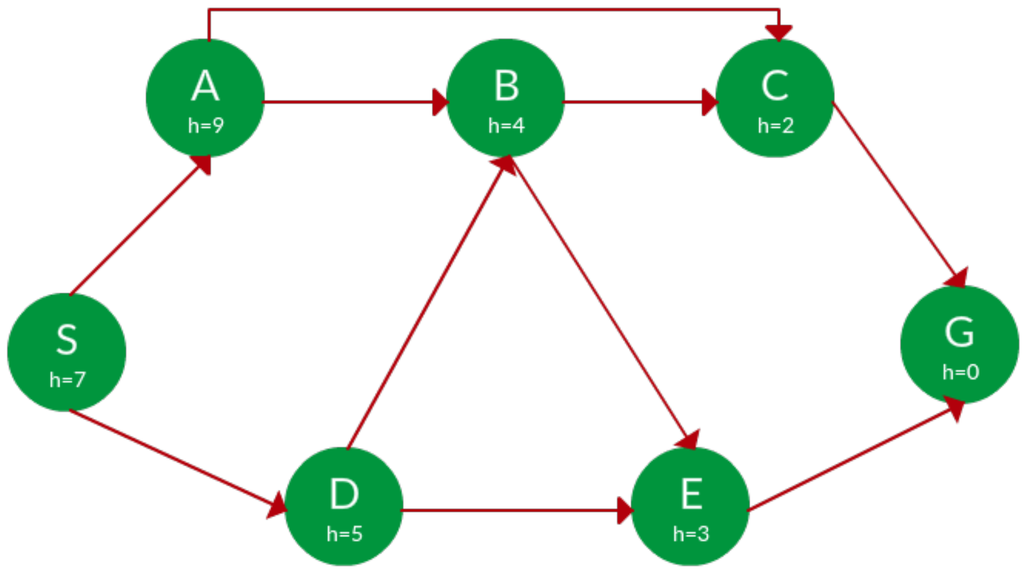


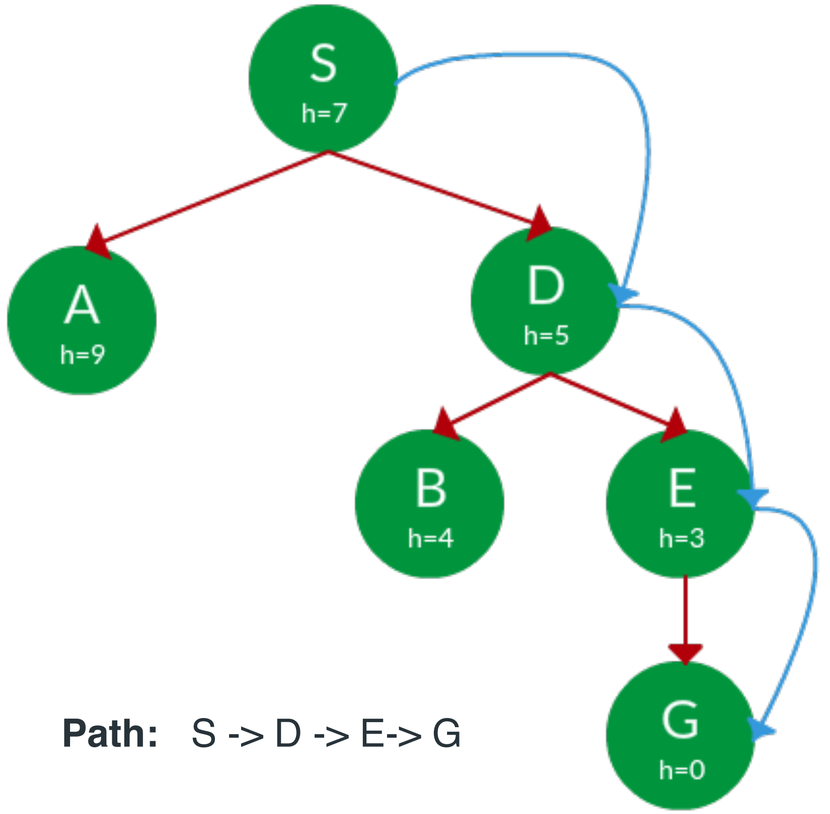
Path: S -> D -> C -> G2

Greedy Search

Informed Search

- Choose the node closest to the goal
- The closest node is estimated by the heuristic (h)





Path: S -> D -> E-> G

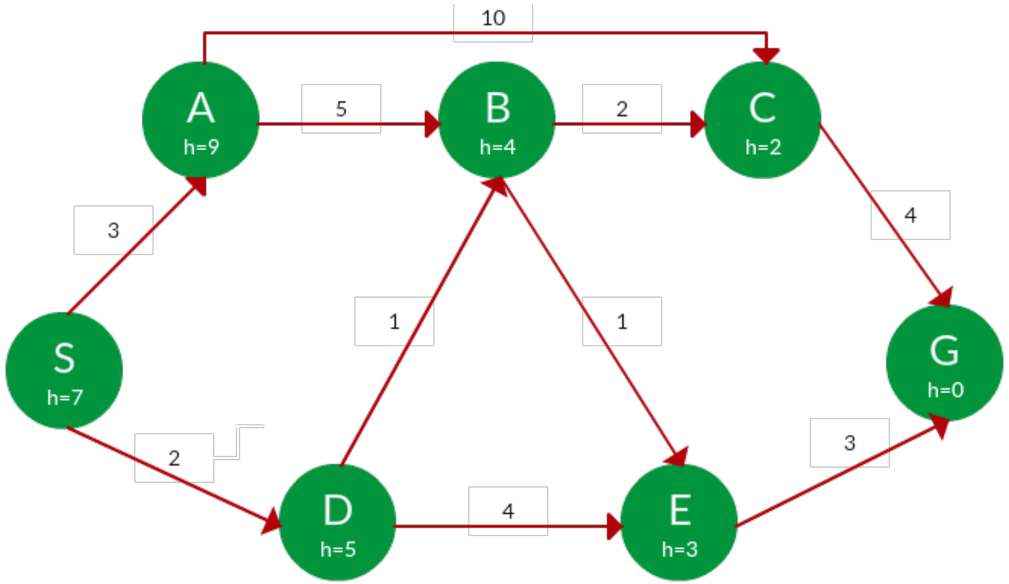
A* Search

Informed Search

- Combines the strengths of uniform-cost search and greedy search
- Heuristic = summation of the cost in UCS + cost in greedy search
- f(x) “**the total cost**” = h(x) “**forward cost**” + g(x) “**backward cost**”
- Choose the node with the lowest f(x) value

A* Search

Informed Search



	h(x)	g(x)	f(x)
S->A **S->D**	9 5	3 2	12 7
S->D->B **S->D->E**	4 3	2+1=3 2+4=6	7 9
S->D->B->C **S->D->B->E**	2 3	2+1+2=5 2+1+1=4	7 7
S->D->B->C->G **S->D->B->E->G**	0 0	2+1+2+4=9 2+1+1+3=7	9 7

Path: S -> D -> B -> E-> G

Cost: 7