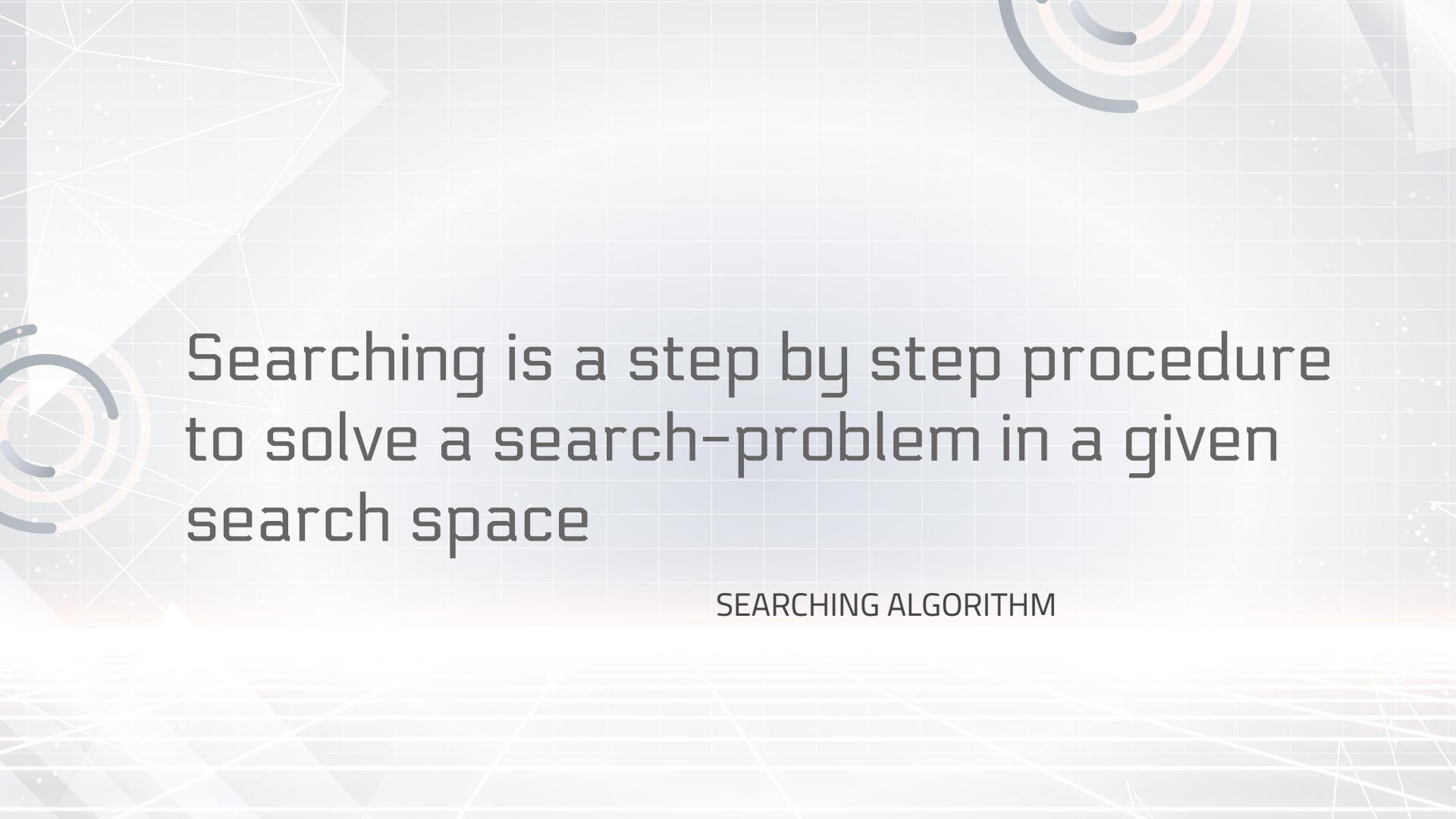




SEARCHING ALGORITHM

RATNA MUFIDAH, S.Kom., M.Kom.



Searching is a step by step procedure
to solve a search-problem in a given
search space

SEARCHING ALGORITHM



BLIND SEARCH/ UNINFORMED SEARCH

Image source:

<https://www.dreamstime.com/illustration/play-blind-man.html>

Blind Searching/ Uninformed Searching

A **blind search** (also known as an **uninformed search**) is one that has no knowledge of the domain. A blind search can only distinguish between non-goal and goal states.

Why would you use a **blind search**?

1. In some cases, there is **no usable initial information**.
2. We will **only know the answer** when **we find it** or **arrive at our destination**.

Blind Searching (Uninformed Searching)

Breadth-First Search (BFS)

Depth-First Search (DFS)

Uniform Cost Search (UCS)

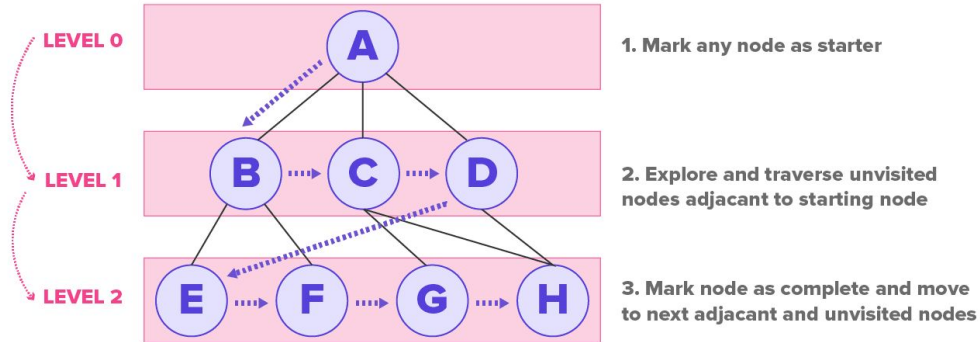
Depth Limited Search (DLS)

Iterative Deeping Depth First Search (IDDFS)

Bidirectional Search (BS)

Breadth-First Search (BFS)

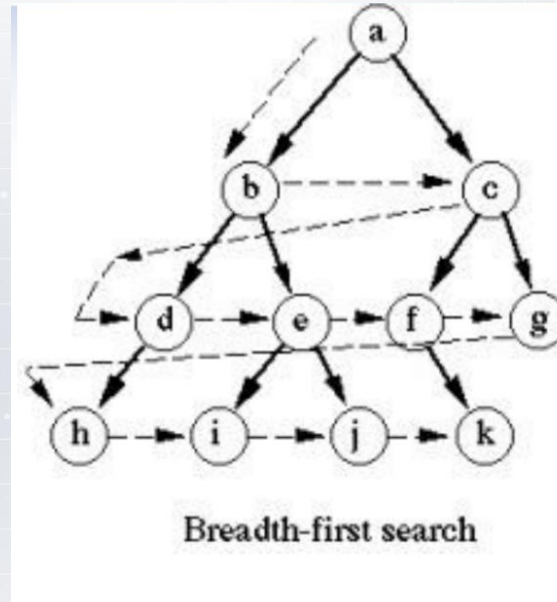
ARCHITECTURE OF BFS



Source: <https://hackr.io/blog/breadth-first-search-algorithm>

Breadth-First Search (BFS)

- Starting at level 0, the root node, A, then proceeds to the destination node (goal), K.
- Using the **First In, First Out principle (FIFO)**.
- Using the **Queuing Principle**.



Breadth-First Search (BFS)

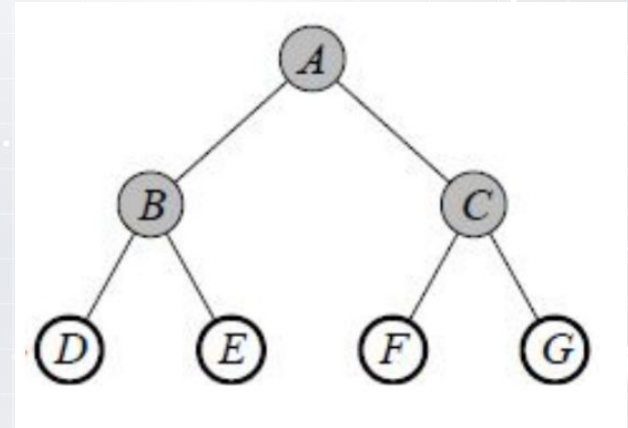
The search is carried out sequentially from left to right on all nodes at the same level.

If no solution is found at one level, the search is continued at the next level ($n+1$).

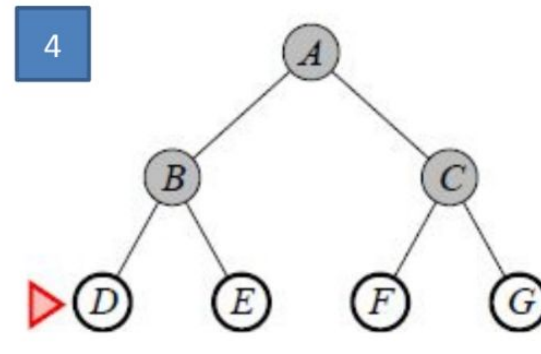
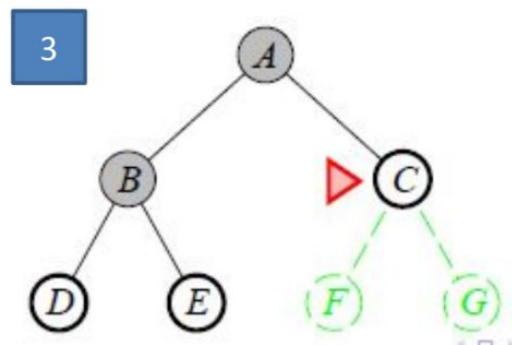
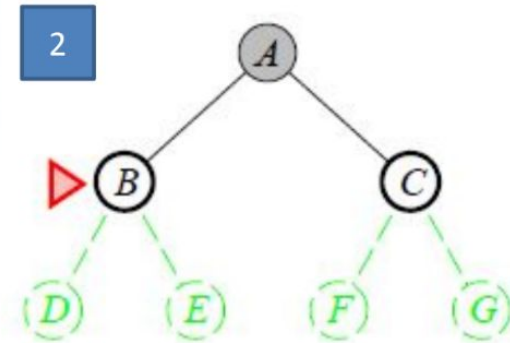
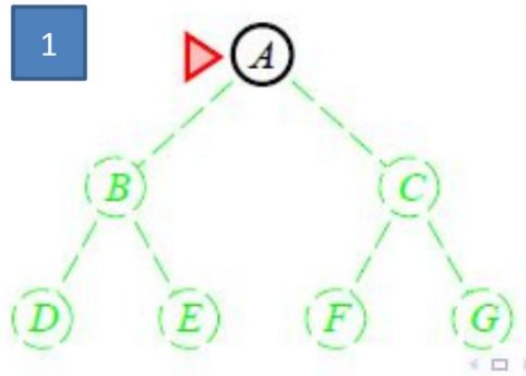
Example:

Root: A

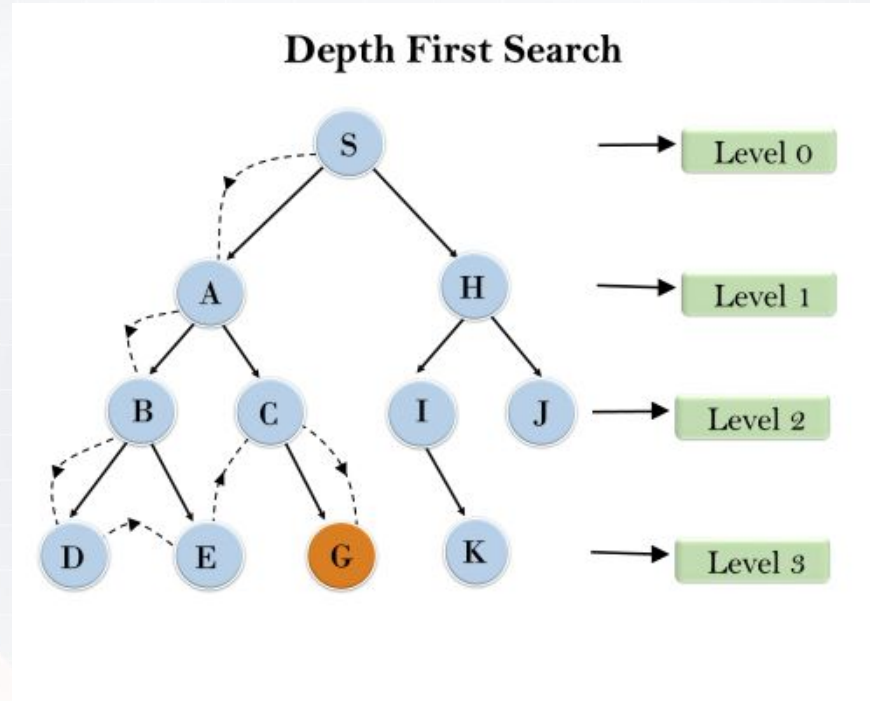
Goal: D



Breadth-First Search (BFS)



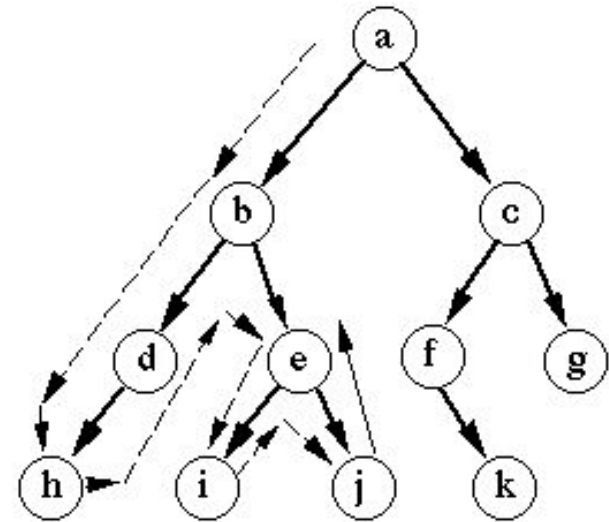
Depth-First Search (DFS)



Source: <https://www.johnwennardfoja.top/ProductDetail.aspx?iid=154589730&pr=42.88>

Depth-First Search (DFS)

- The search is performed on a node at each level, beginning at the far left.
- If the solution is not found at the deepest level, the search moves to the right end, while the left end is erased from memory.
- If a solution is not found at the deepest level, the search proceeds to the previous level.



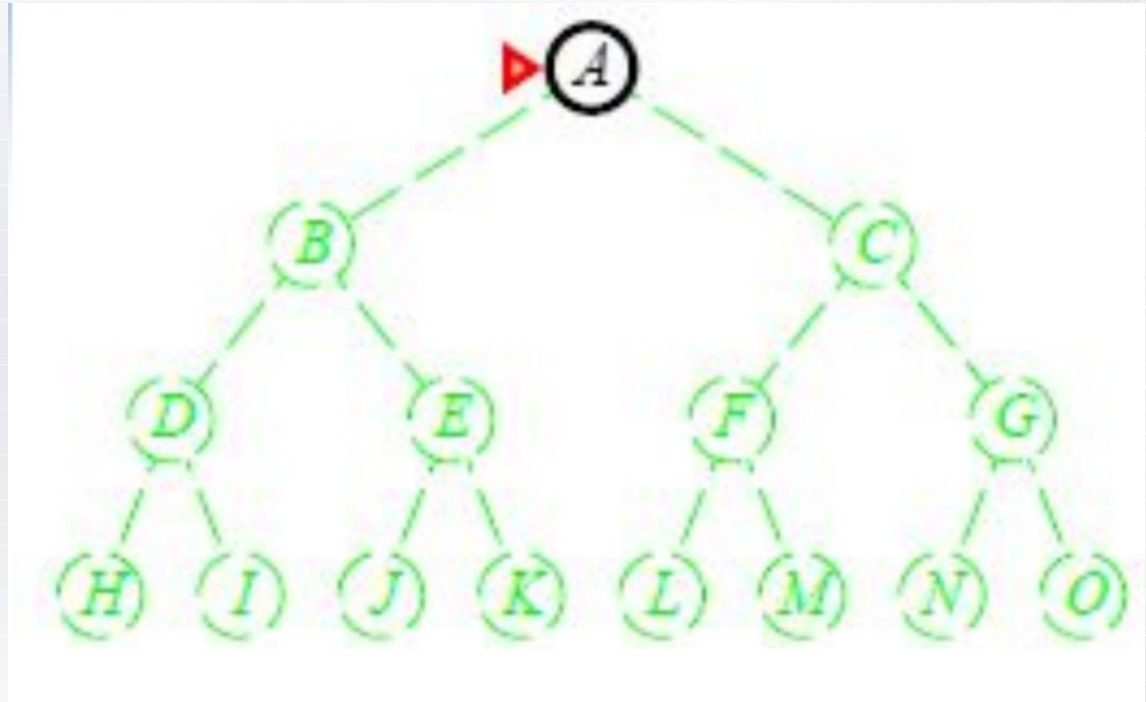
Depth-first search

Depth-First Search (DFS)

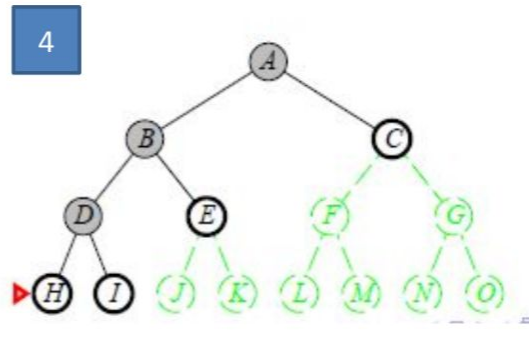
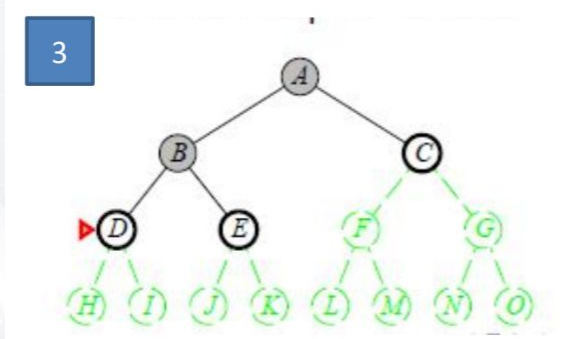
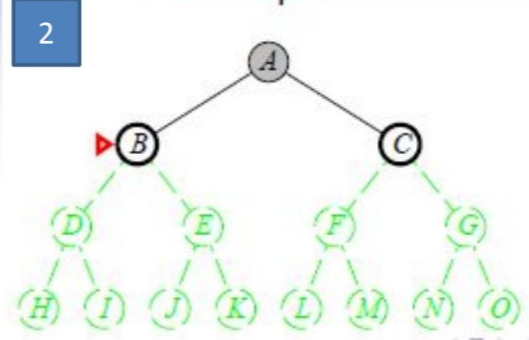
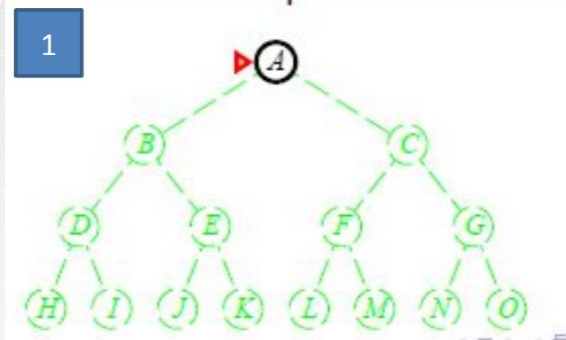
Example:

Root: A

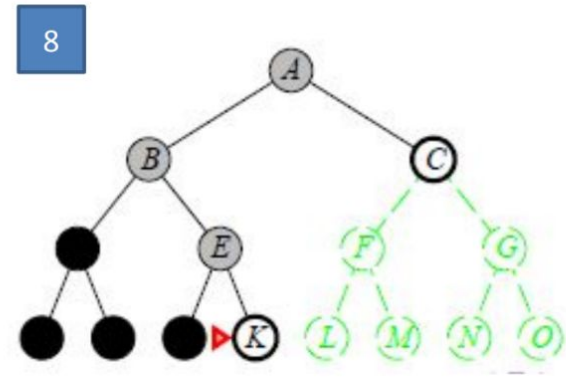
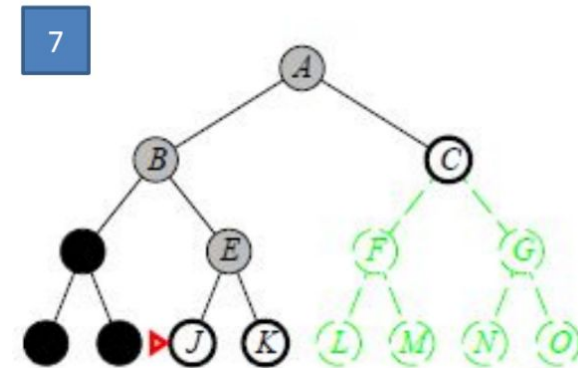
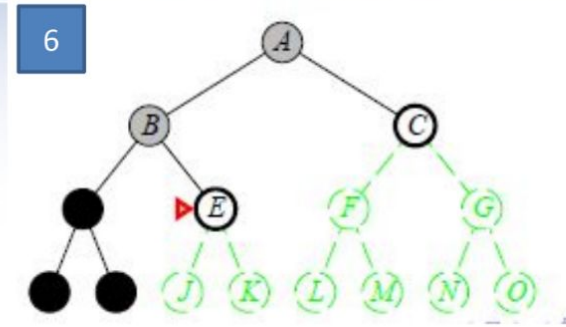
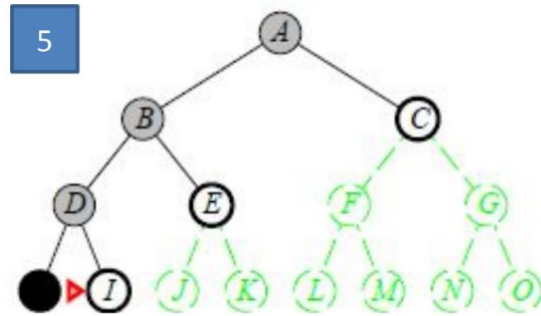
Goal: M



Depth-First Search (DFS)



Depth-First Search (DFS)



Depth-First Search (DFS)

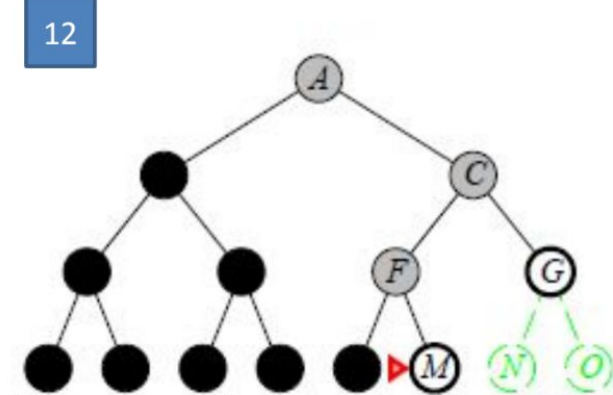
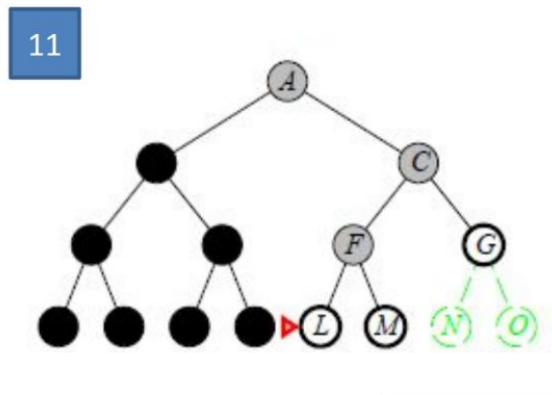
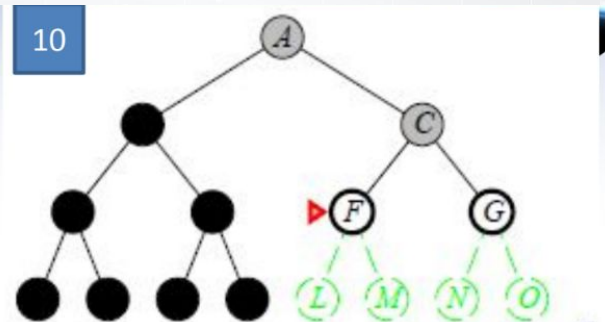
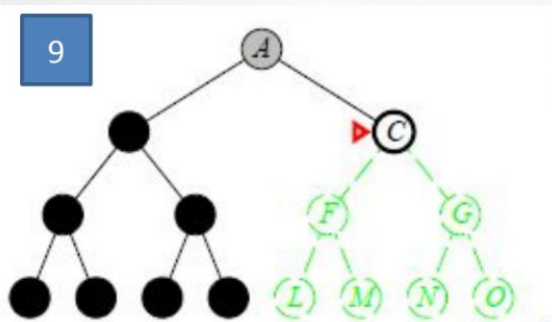




Image source:
<https://hypersense.subex.com/aiglossary/heuristic/>

HEURISTIC SEARCH

Heuristic Search

Blind searches have these **drawbacks**:

- Long access time
- A lot of memory is required.
- Because the problem space is large, it is not suitable for computers with limited speed and memory.

Solution

Heuristic Search

The **heuristic function** is used to assess the circumstances of individual problems and determine how far they can be used to achieve the desired result.

This method employs a heuristic function, which computes the estimated cost from one node to the next.

Heuristic Search

Generate and Test

Hill Climbing

Best First Search

Alpha Beta Pruning

Simulated Annealing

Min-Max

Local Search Algorithms

Local Beam Search

Best First Search

- Combination of DFS and BFS methods
- If the node on a higher level has a lower heuristic value, **the best-first search** allows it to visit a lower node.
- Algorithms are classified into two types:
 - **Greedy's Best-First Search**
estimated cost $f(n) = h(n)$
 - **A***
 $f(n) = g(n) + h(n)$

Greedy Best-First Search



Greedy best-first search algorithm always selects the path which appears best at that moment.

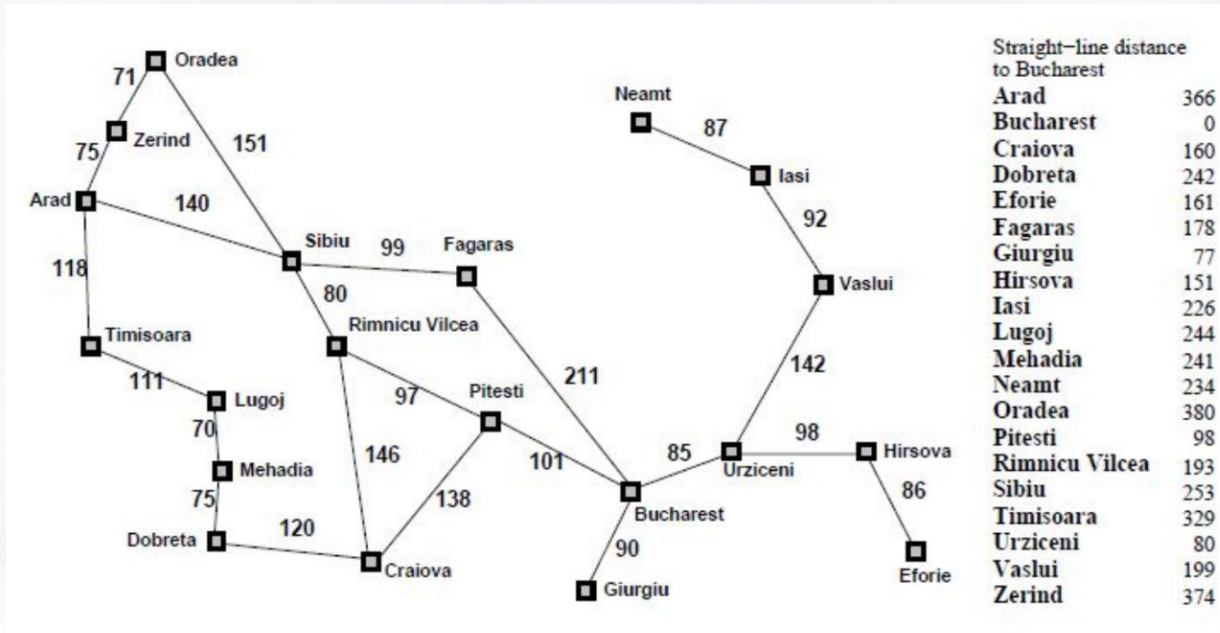
estimated cost $f(n) = h(n)$

$h(n)$ = estimated cost from node n to the goal



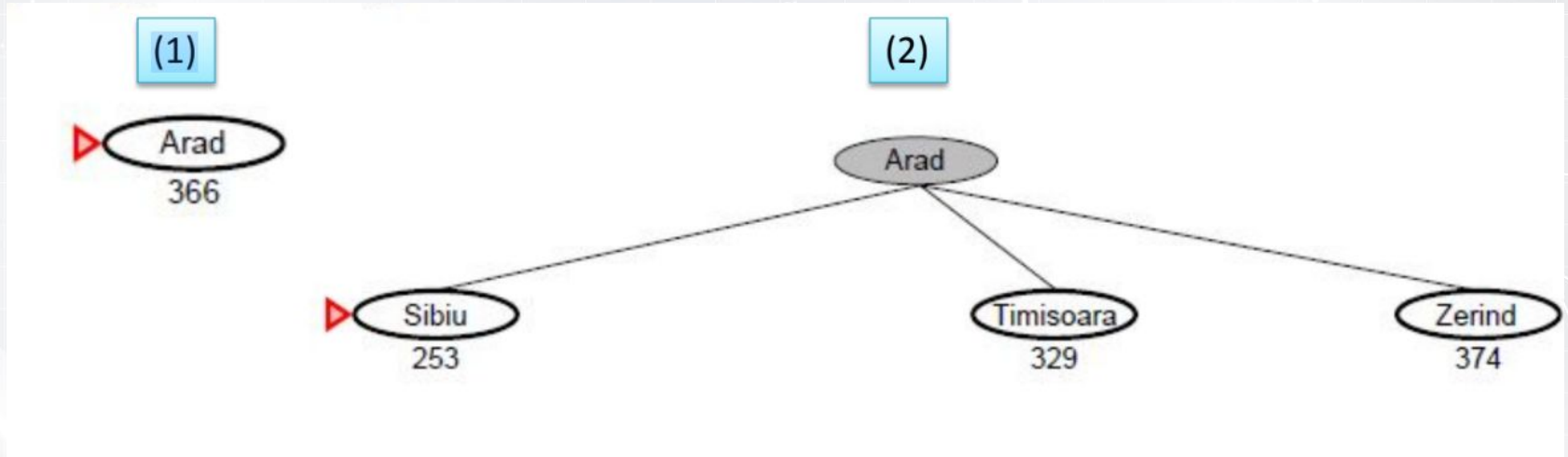
Greedy Best-First Search

Case: Find the shortest route from Arad to Bucharest.



Greedy Best-First Search

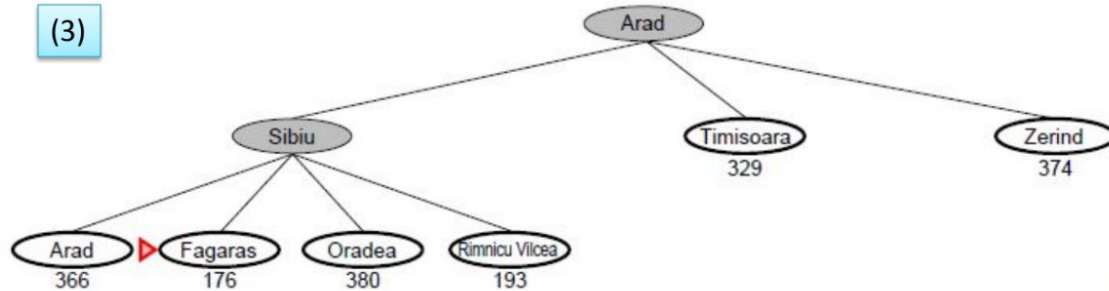
Case: Find the shortest route from Arad to Bucharest.



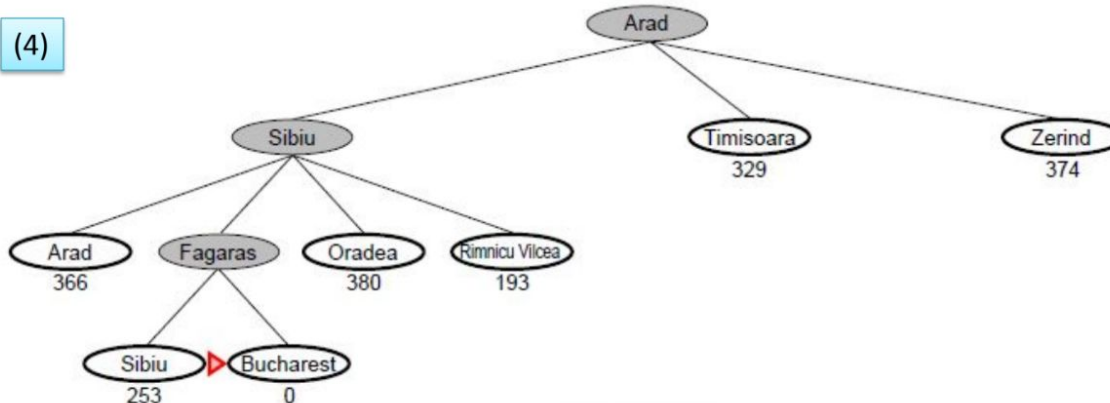
Greedy Best-First Search

Case: Find the shortest route from Arad to Bucarest.

(3)

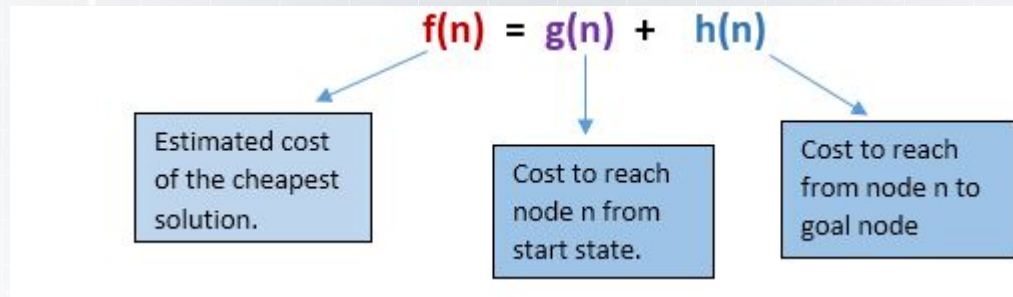


(4)



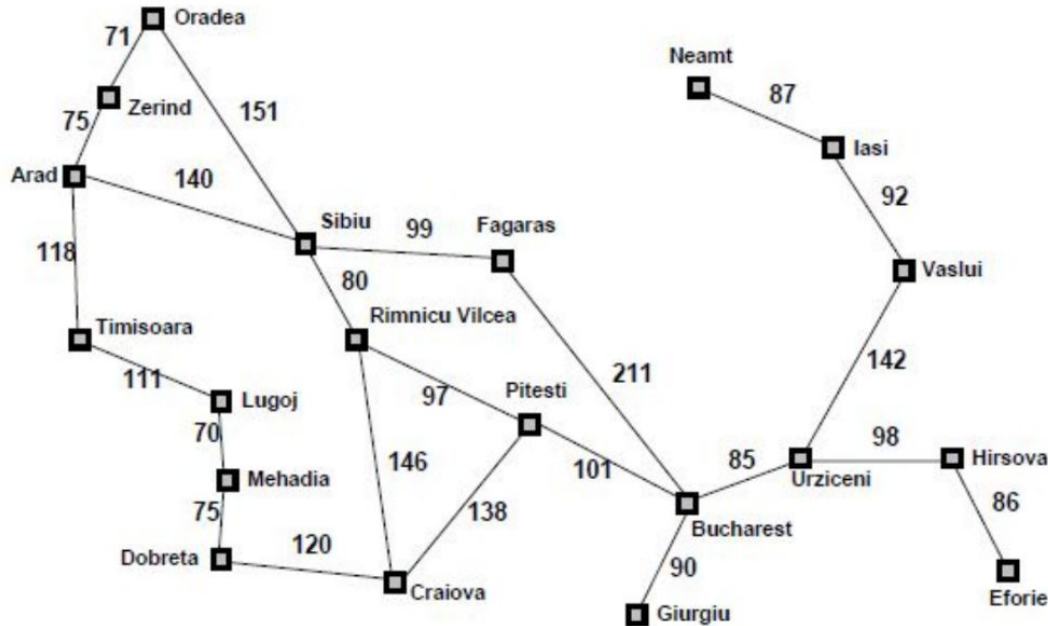
A* Best-First Search

A* search is the most commonly known form of best-first search. It uses heuristic function $h(n)$, and cost to reach the node n from the start state $g(n)$.



A* Search

Case: Find the shortest route from Arad to Bucharest.

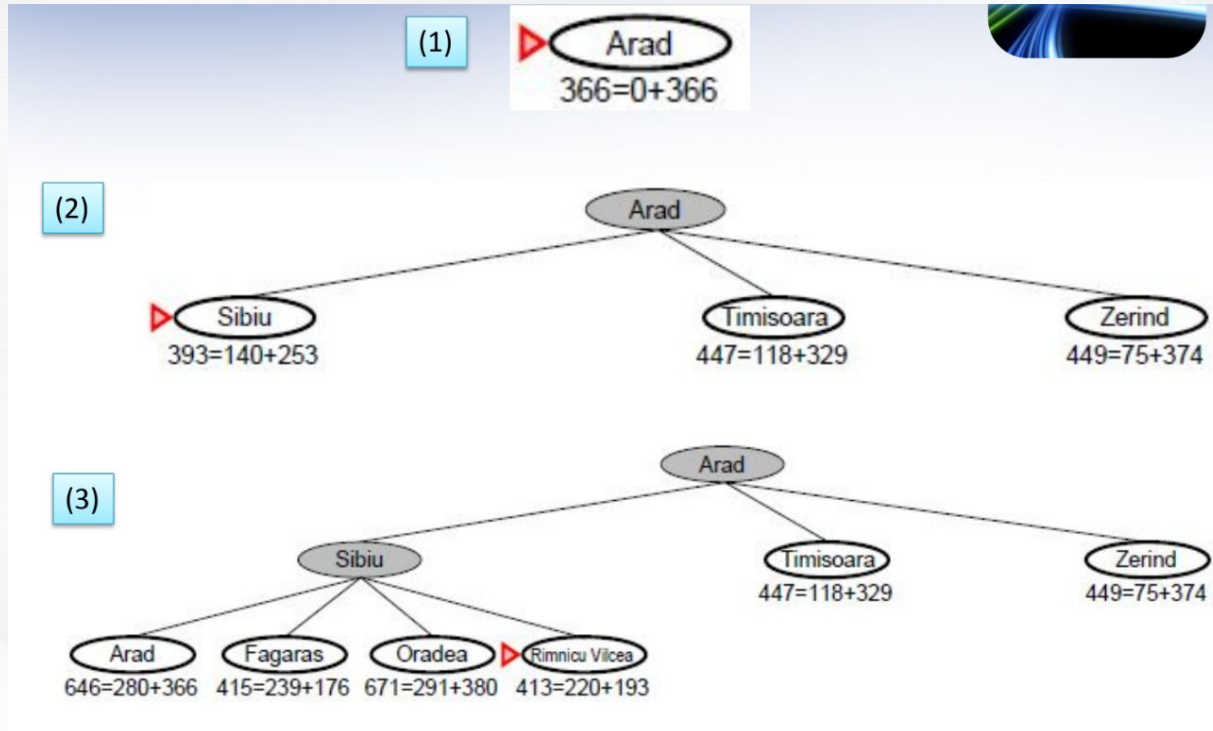


Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

A* Search

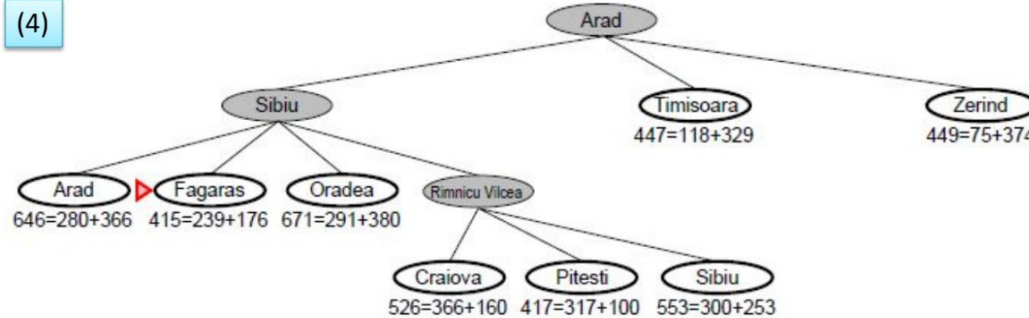
Case: Find the shortest route from Arad to Bucarest.



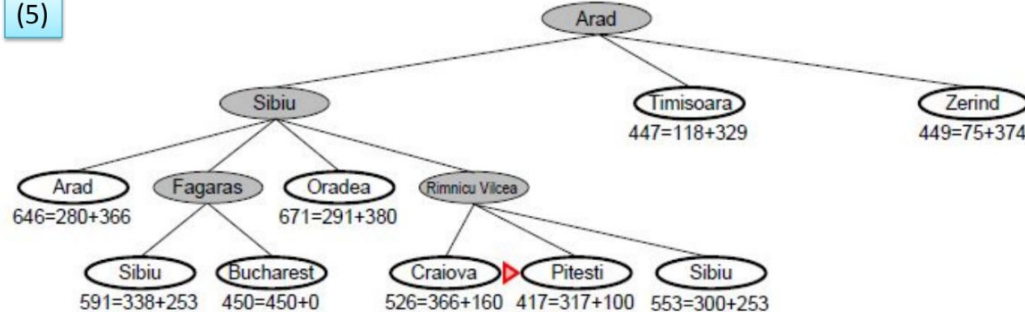
A* Search

Case: Find the shortest route from Arad to Bucharest.

(4)



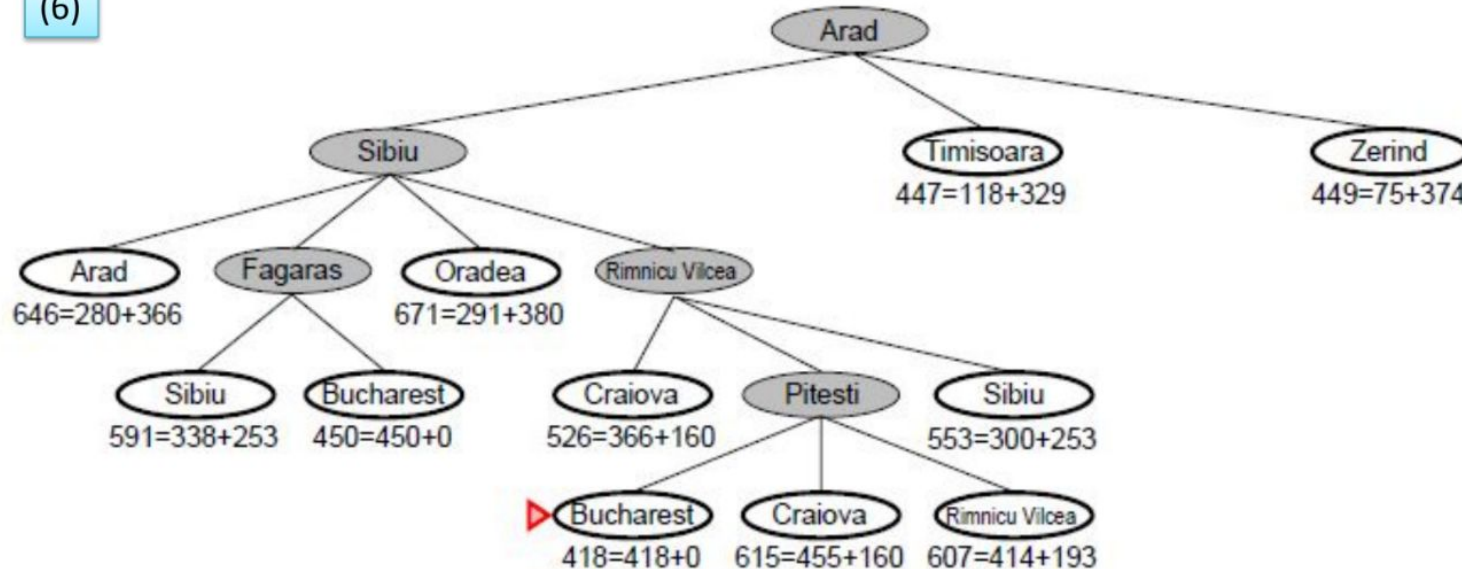
(5)



A* Search

Case: Find the shortest route from Arad to Bucarest.

(6)





THANKS!

See You

CREDITS: This presentation template was created by
Slidesgo, including icons by **Flaticon** and infographics
& images by **Freepik**

