Uninformed Search

N Geetha
A M & C S
P S G College of Technology

Search

- Search problem is associated with
  - What to search ? Key (goal)
  - Where to search ? Search space
- Solving a search problem has 2 phases
  - Generation of the state space
  - Searching the desired problem state (Goal) in that space
  - The state space is expanded in steps as it may cause a significant blockage of space
  - The goal is searched after each incremental expansion of the state space

Search

- Sometimes state space may be infinite
- Most search problems suffer from the problem of combinatorial explosion
- Search space is the narrowed down version of the state space

Example

1	2	8
	4	3
7	6	5

initial state

1	2	3
8		4
7	6	5

goal state

Uninformed search

- Brute Force Search
- Depth-first Search
- Breadth-first Search
- Bidirectional Search
- Iterative Deepening Search

Uninformed Search

- **Brute Force Search (Generate and Test)**
- Select an appropriate operator applicable to state and obtain the new state
- Initial and goal states are known

- Algorithm:

1. state = Initial_state
2. While state <> goal_state do

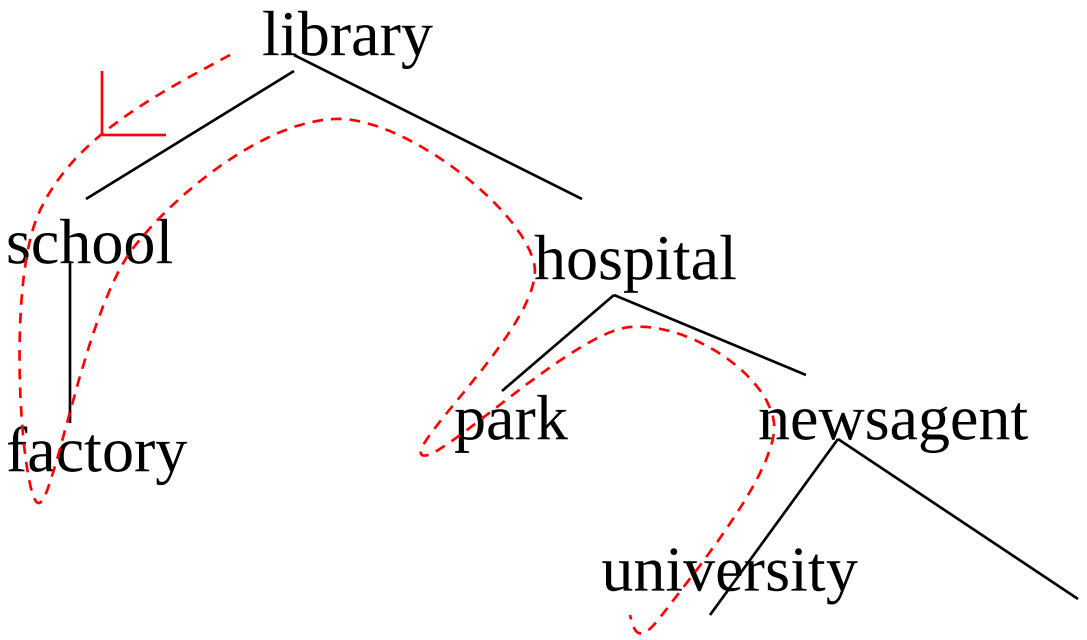
   state = operator(state)
3. End.

Depth-First Search

- **Procedure Depth-First Search (Vertical Search / Backtracking)**
- /* OPEN is a stack data structure; CLOSED is a
- S1. Put the initial state in OPEN
- S2. LOOP: If (empty(OPEN)), then exit(failure).
- /* If all the nodes have been examined and the search terminates, the result of the search is taken to be a failure */
- S3. n = first(OPEN). /* fetch initial state from OPEN*/
- S4. if goal(n), then exit(success).
- /* if n is the goal state, the search has succeeded and is terminated */
- S5. remove(n, OPEN); add(n, CLOSED)
- /* n is removed from OPEN and is added to CLOSED */
- S6. Expand n; Put all the child nodes of n at the head of OPEN and attach links from the child nodes to n (child nodes which do not appear in OPEN or CLOSED are alone put at the head of OPEN)
- S7. Go to S2.

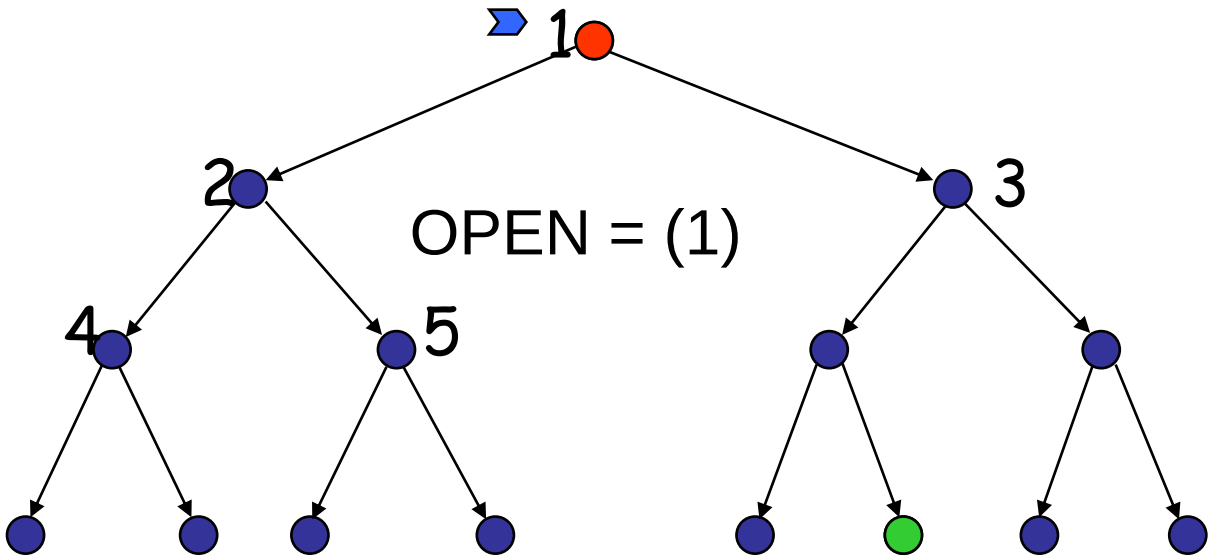
Depth first search

- Nodes explored in order: library, school, factory, hospital, park, newsagent, university.



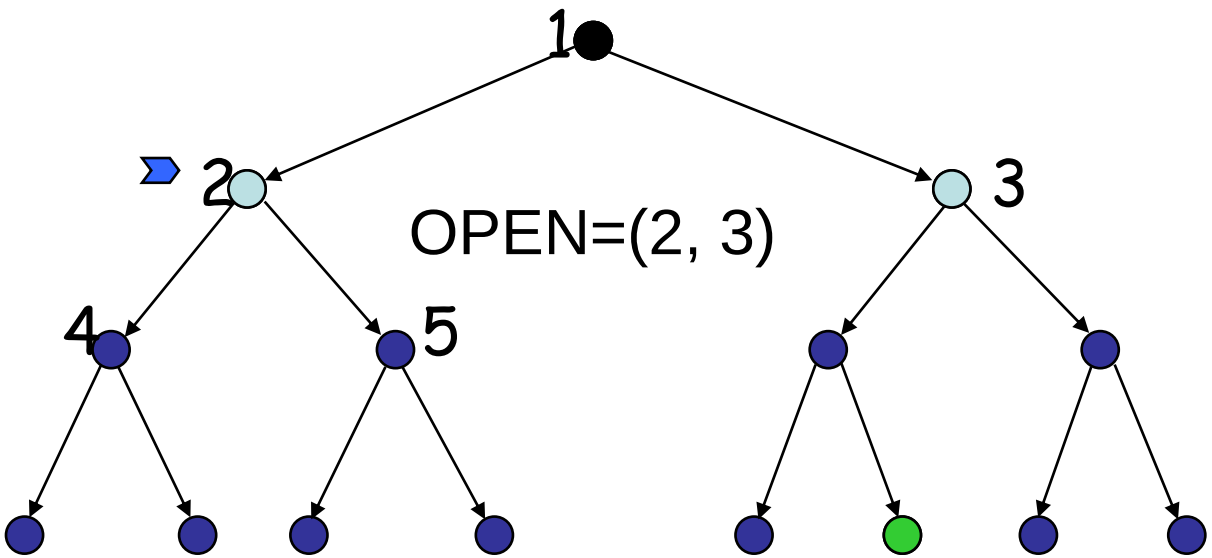
Depth-First Strategy

- Example 1:



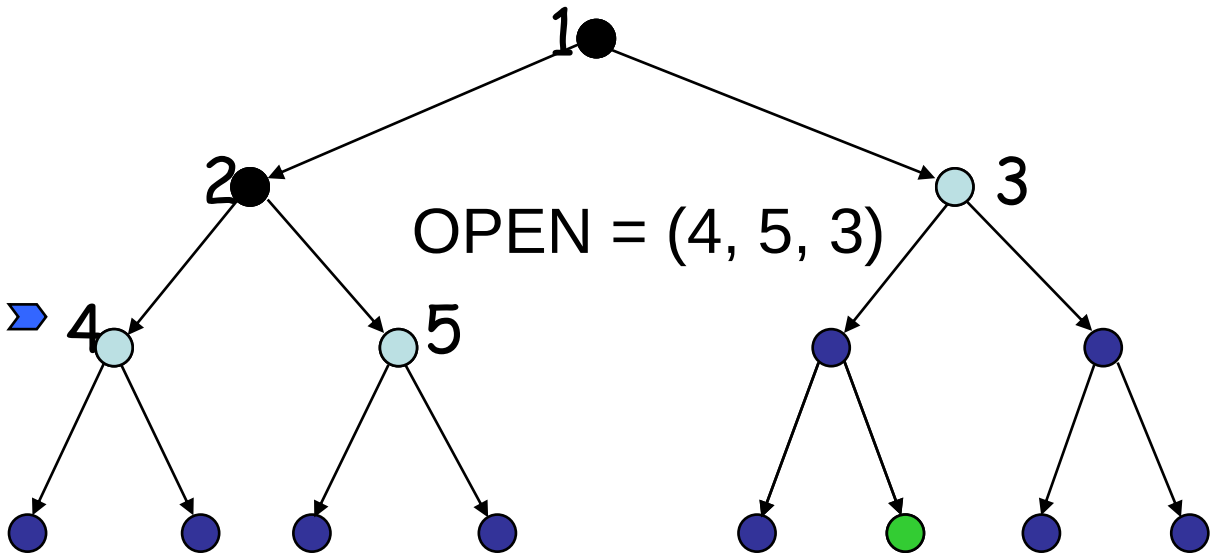
Depth-First Strategy

- Example 1:



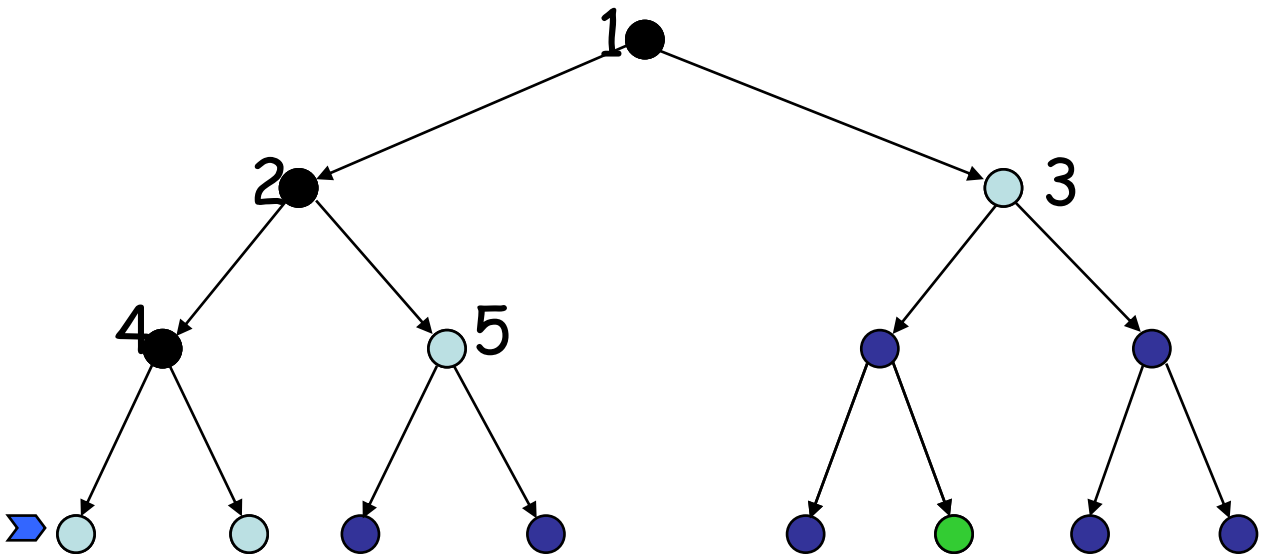
Depth-First Strategy

- Example 1:



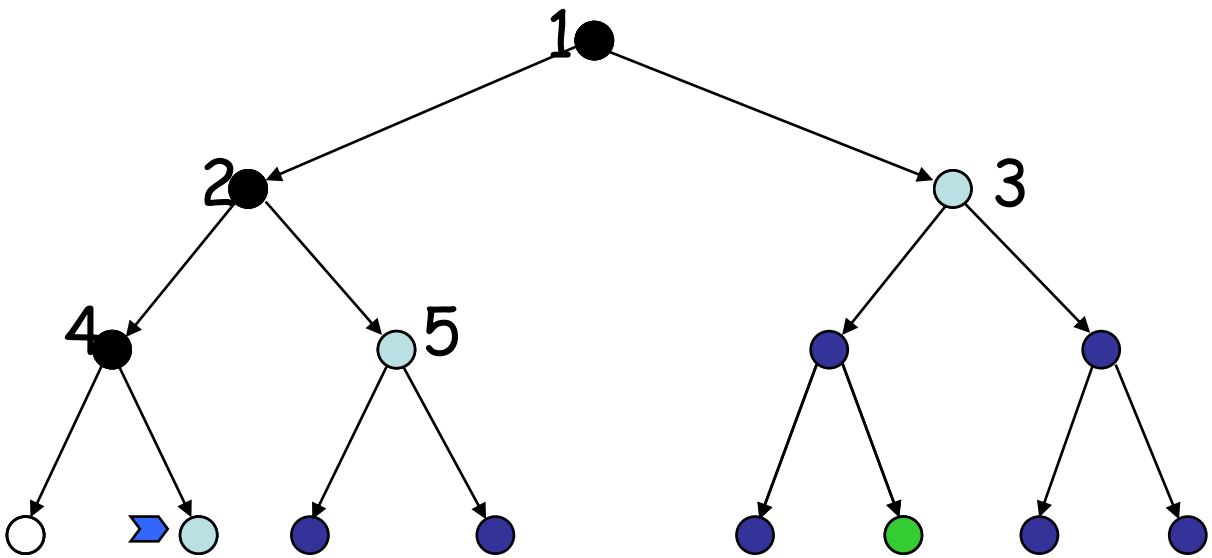
Depth-First Strategy

- Example 1:



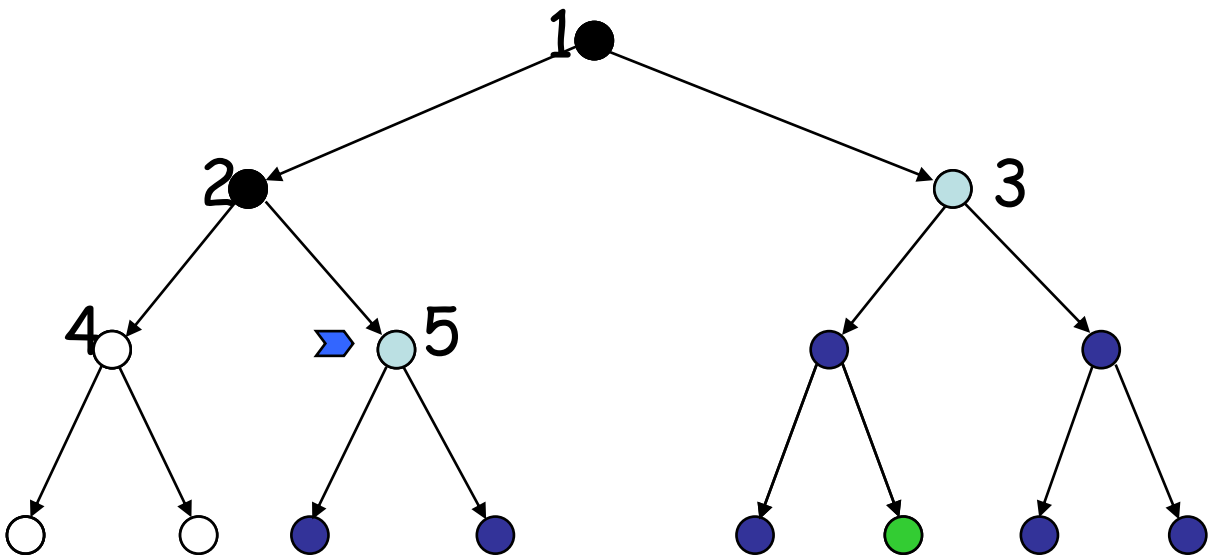
Depth-First Strategy

- Example 1:



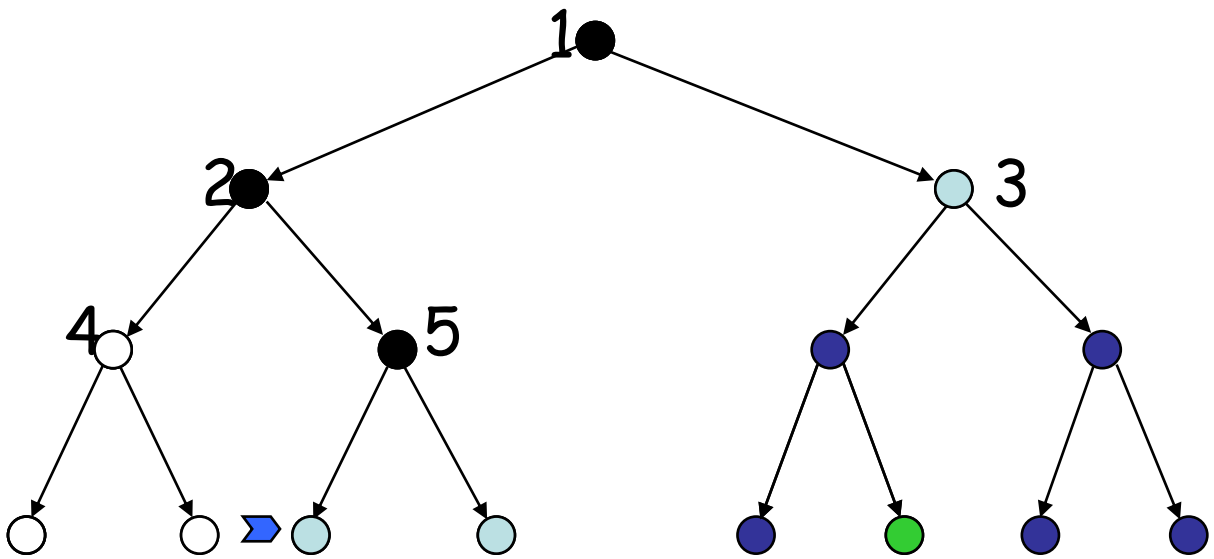
Depth-First Strategy

- Example 1:



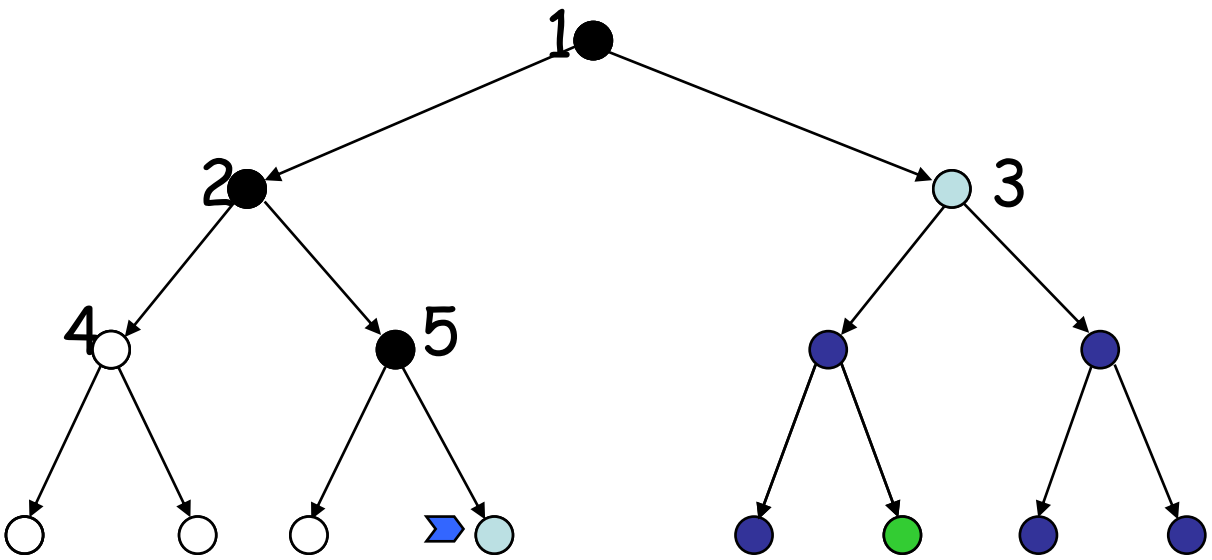
Depth-First Strategy

- Example 1:



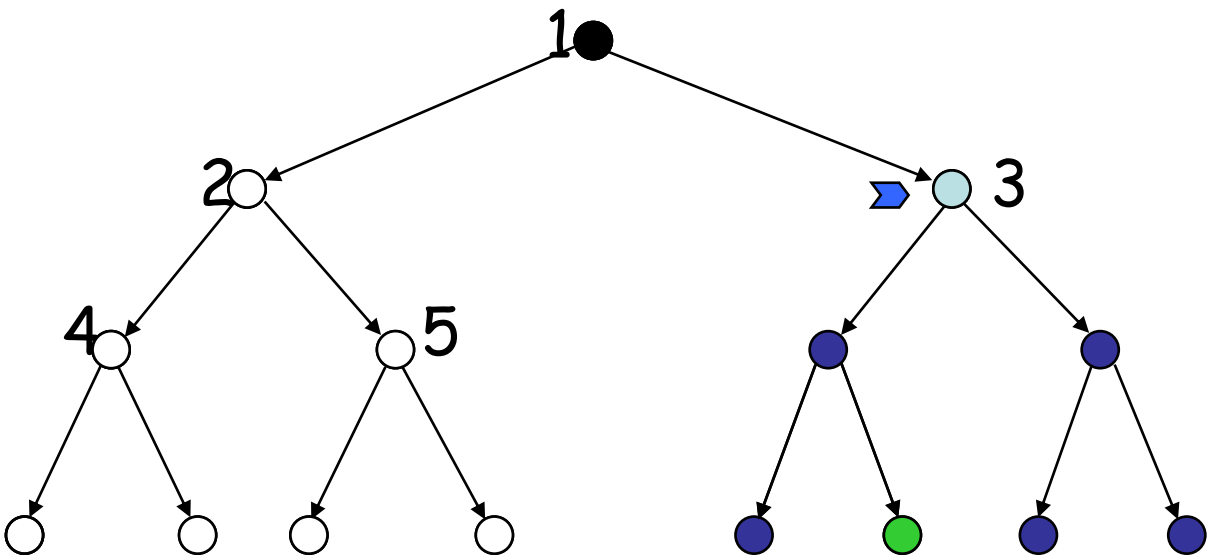
Depth-First Strategy

- Example 1:



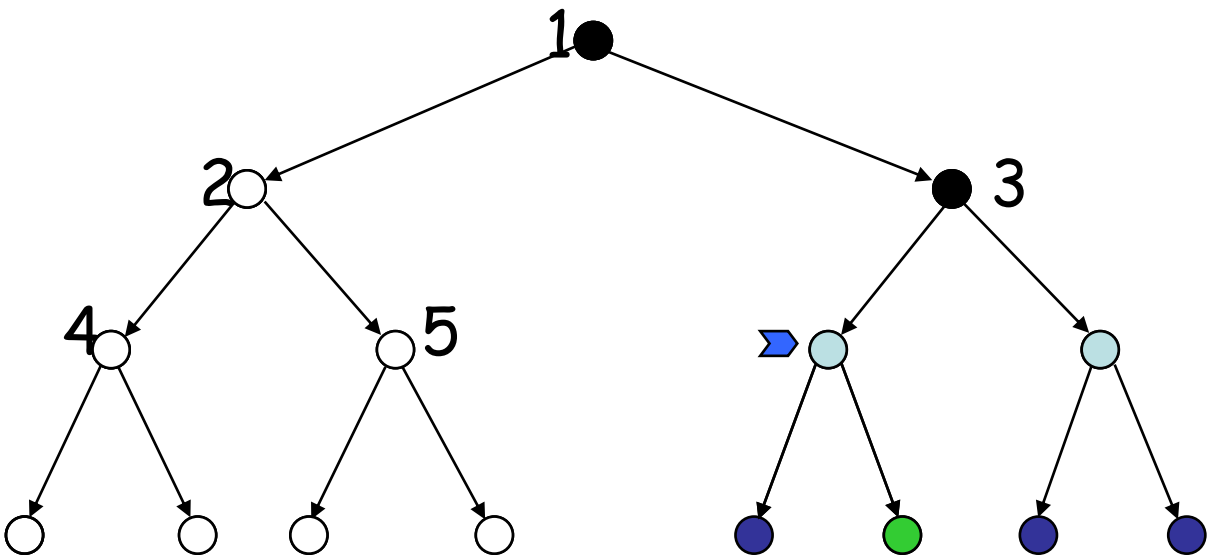
Depth-First Strategy

- Example 1:



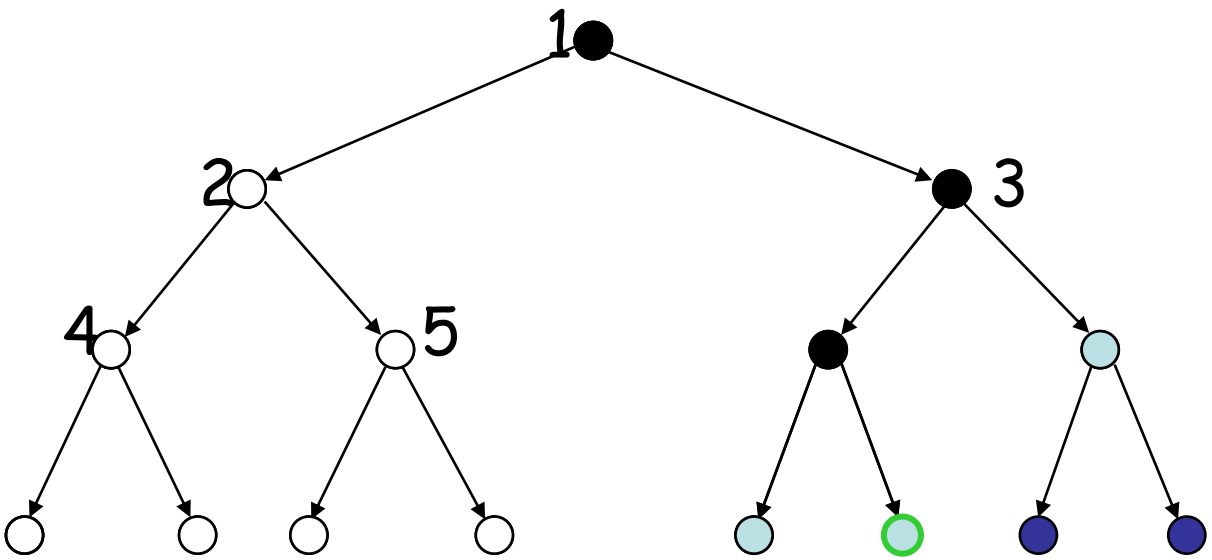
Depth-First Strategy

- Example 1:



Depth-First Strategy

- Example 1:



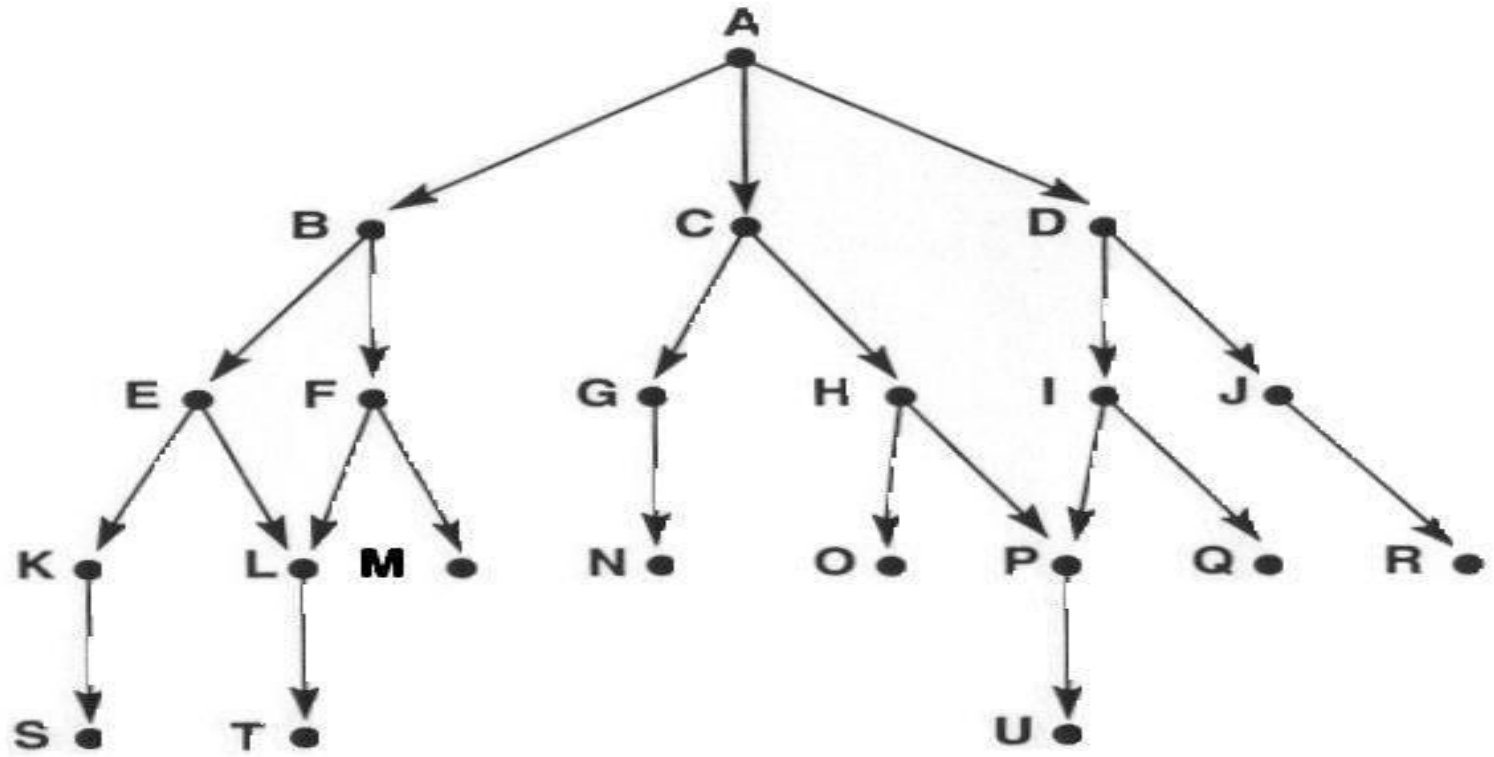
Example 1

n	OPEN	CLOSED	Remarks
	1		Initial
1	~~1~~	1	Goal(1) is false
1	2,3	1	Expand 1
2	~~2~~,3	1,2	Goal(2) is false
2	4,5,3	1,2	Expand 2
4	~~4~~,5,3	1,2,4	Goal(4) is false
4	8,9,5,3	1,2,4	Expand 4
			Goal(12) is true

Evaluation Measures

- b: branching factor
- d: depth of shallowest goal node
- m: depth of the search tree
- Depth-first search is
  - Not complete
  - Not optimal
- Best case
  - Time and space complexity is O(d) – extremely fast
- Average case
  - $1 + b + b^2 + \ldots + b^d = (b^{d+1}-1)/(b-1) = O(b^d)$
  - Time complexity is : $O(b^d)$
  - In practice, DFS is usually quicker than BFS
- Worst case
  - Time complexity is $O(b^m)$ – slow
  - Space complexity is O(b*m) – excellent

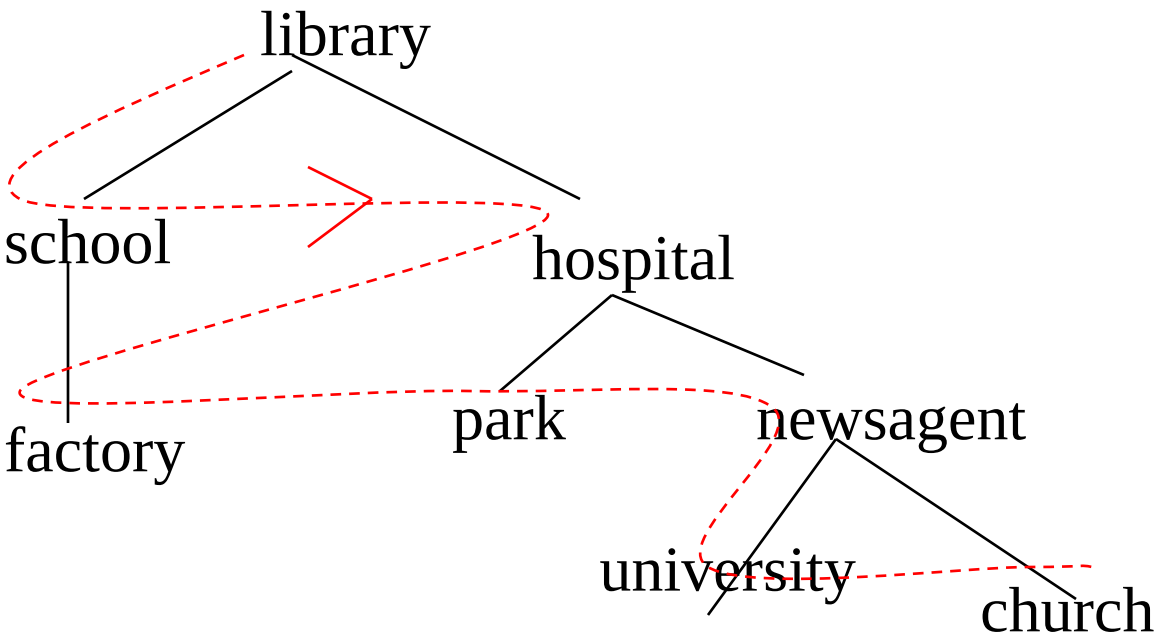
Example 2 : A:initial, U:goal



Graph for breadth- and depth-first search examples.

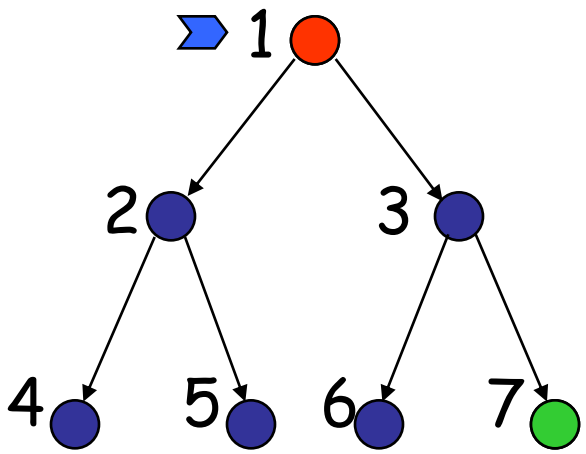
Breadth first search

Explore *nodes* in tree order: library, school, hospital, factory, park, newsagent, uni, church.
(conventionally explore left to right at each level)



Breadth-First Strategy

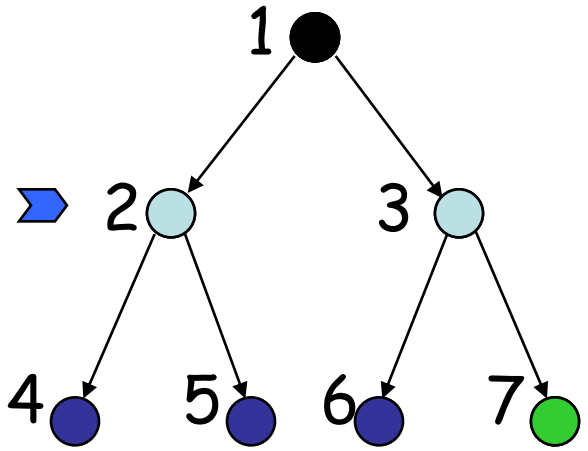
- Example :1



OPEN = (1)

Breadth-First Strategy

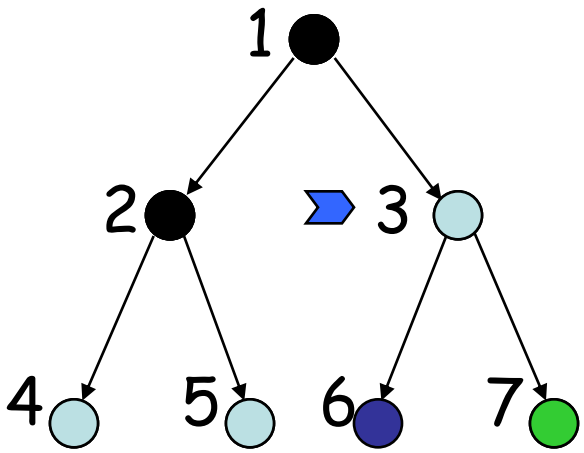
- Example :1



OPEN = (2, 3)

Breadth-First Strategy

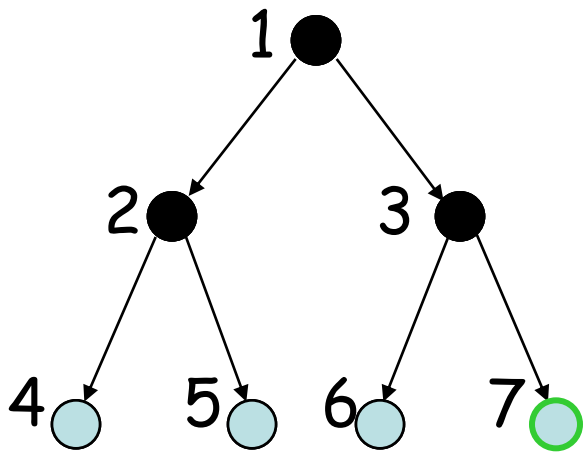
- Example :1



OPEN = (3, 4, 5)

Breadth-First Strategy

- Example :1



OPEN = (4, 5, 6, 7)

References

- Elaine Rich, Kevin Knight, “Artificial Intelligence”, Tata Mcgraw Hill, 2002.
- Amit Konar, ”Artificial Intelligence and Soft Computing”, CRC Press, 2000.
- Peter Norvig, Stuart Russel, “Artificial Intelligence: A modern Approach”, Prentice Hall of India, 2006.
- Florin Leon, http://eureka.cs.tuiasi.ro/~fleon