Artificial Intelligence

Lab 4

Uninformed Search

Agenda

What is Heuristic Search?

Informed vs. Uninformed Search

Uninformed Search

- Breadth First Search (BFS)
- Depth First Search (DFS)
- Uniform Cost Search (UCS)

Milestone 2

Searching Techniques

- There are many problems that require searching for an answer within a domain.

- There are some single-player games such as Tic Toc Toe, Sudoku, crossword, etc. The search algorithms help you to search for a particular position/move in such games.

- There are many possible solutions to a given problem and we do not know which ones are correct.

- Also, going through every single solution is not possible because it is prohibitively expensive.

Search Cont.

- In such cases, we rely on a rule of thumb that we develop based on our past experiences that helps us narrow down the search by eliminating the options that are obviously wrong.
- This rule of thumb is called a **heuristic**. The method of using heuristics to guide our search is called **heuristic search**.
- **Heuristic search** will help us to order those options so that we are more likely to get to the better solutions first.
- Heuristic search uses information about the problem to help directing the path through the search space.

Informed Search (Heuristic Search)

- Heuristic search, is called **Informed search** because it uses **prior information** or rules to eliminate unnecessary paths.

- For example, In the graph problem at each node, we can define a heuristic function that returns a score that represents the estimate of the cost of the path from the current node to the goal.

Informed Search (Heuristic Search)

- By defining this heuristic function, we are informing the search technique about the right direction to reach the goal. This will allow the algorithm to identify which neighbor will lead to the goal.

- We need to note that heuristic search might not always find the most optimal solution. This is because we are not exploring every single possibility and we are relying on a heuristic.

Uninformed Search (*Blind Search/Brute Force*)

- They do not use any prior information or rules to eliminate some paths.

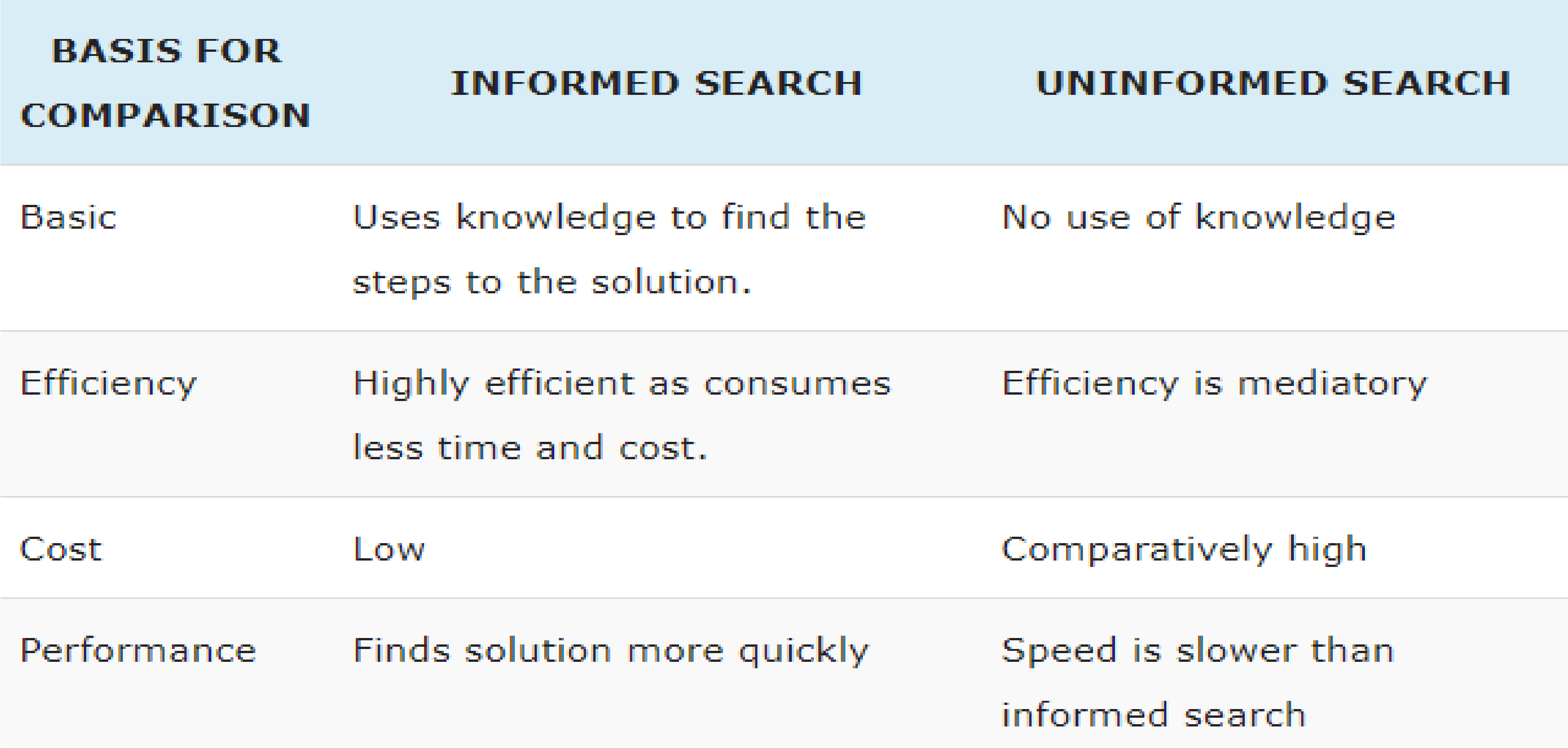
- They check all the possible paths and pick the optimal one.

Informed vs Un-Informed Search

- The prior difference between informed and uninformed search is that the informed search provides the guidance on where and how to find the solution.

- Conversely, the uninformed search gives no additional information about the problem except its specification.

Informed vs Un-Informed Search

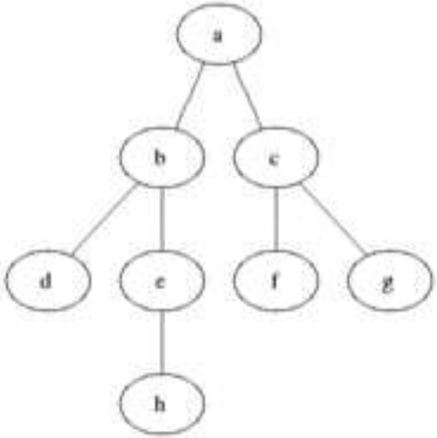


BASIS FOR COMPARISON	INFORMED SEARCH	UNINFORMED SEARCH
Basic	Uses knowledge to find the steps to the solution.	No use of knowledge
Efficiency	Highly efficient as consumes less time and cost.	Efficiency is mediatory
Cost	Low	Comparatively high
Performance	Finds solution more quickly	Speed is slower than informed search

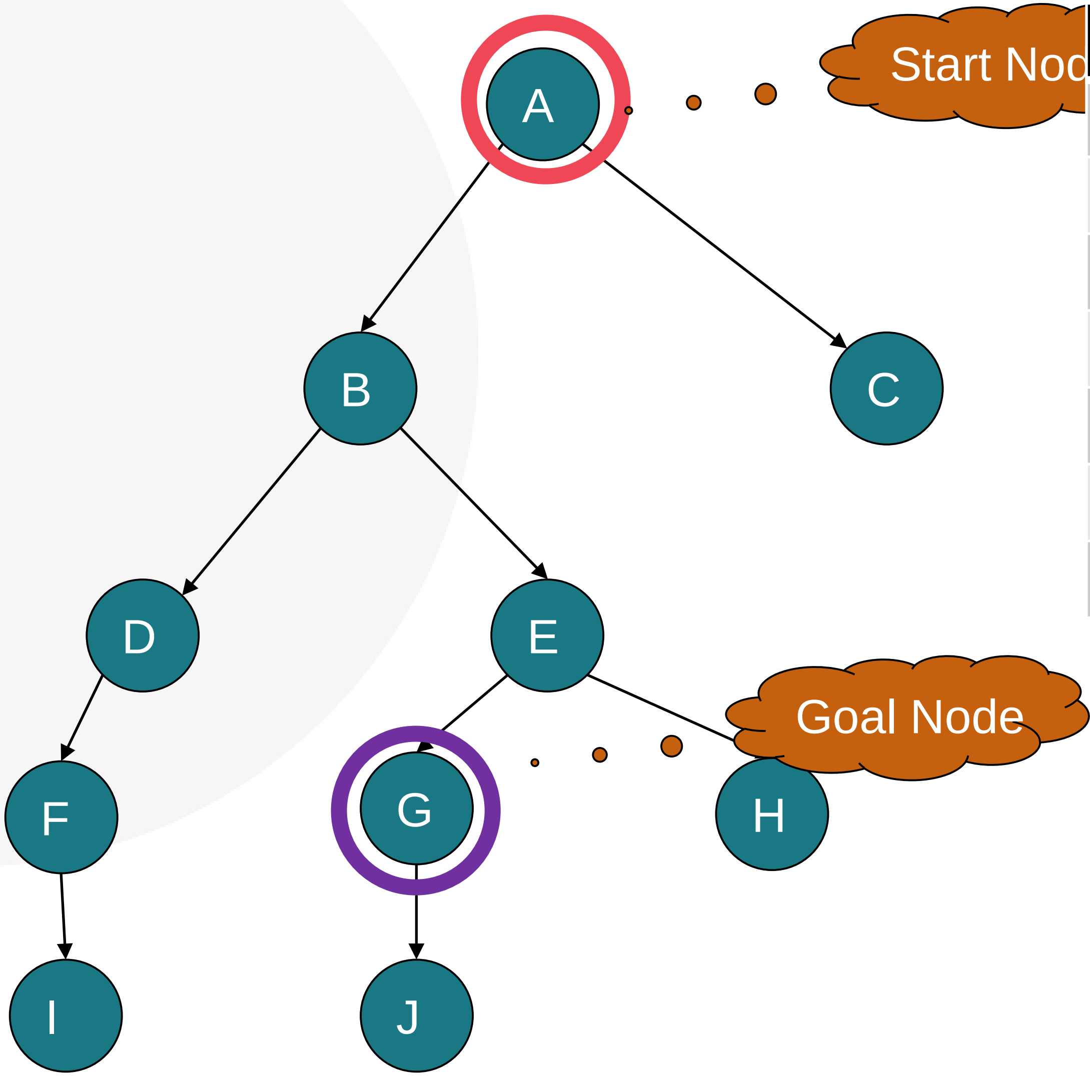
Uninformed Search – BFS – Using Queue

Algorithm Breadth:

```
1.    Initialize two lists OPEN ← {$S_i$}, CLOSED ← {}
2.    If OPEN = {} Then
        return Fail
3. Remove first node S from OPEN and insert it in CLOSED
4. Expand node S Then
    4.1.    Mark all direct successors $S_j$ of node S
    4.2.    For each successor $S_j$ of S do
        4.2.1.  If $S_j$ is final state Then
                    The Solution is ($S_j$, $S_{j-1}$, .., $S_i$)
                .   return SUCCESS
        4.2.2.  Else
                    Insert $S_j$ in OPEN
5.    repeat from step 2
```



Uninformed Search - BFS



Opened	Closed
{A}	{}
{B,C}	{A}
{C, D, E}	{A, B}
{D, E}	{A, B,C}
{E, F}	{A, B,C, D}
{F, G, H}	{A, B,C, D, E}
{G, H, I}	{A, B,C, D, E, F}

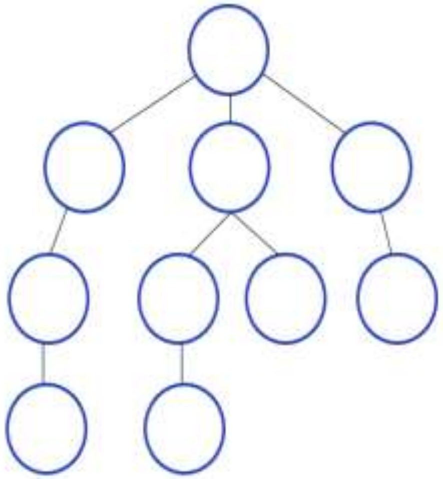
Traversed Path:

{A, B, C, D, E, F, G}

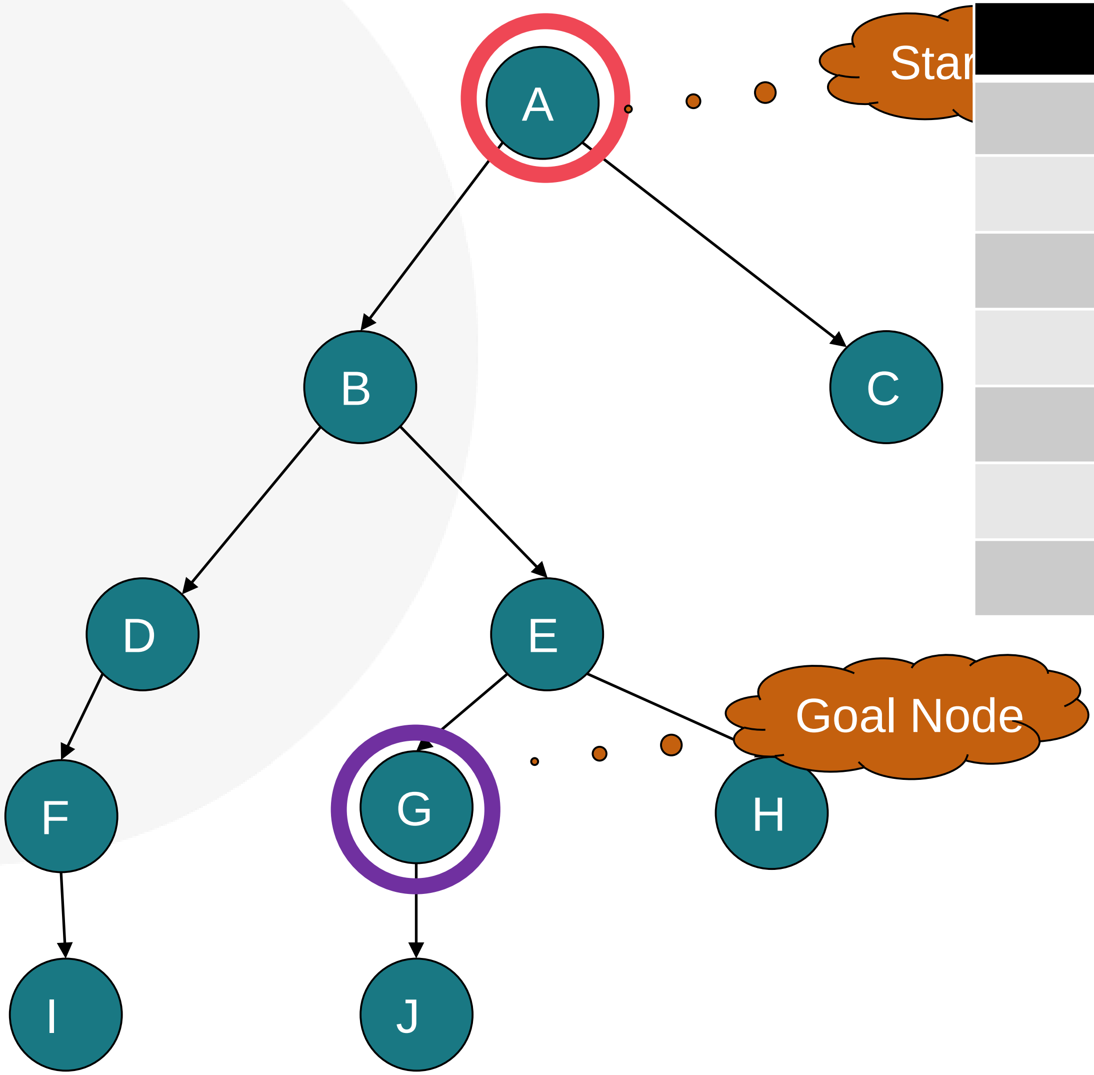
Uninformed Search – DFS – Using Stack

Algorithm Depth:

```
1.      Initialize two lists Visited ← {}, Not-Visited ← {$S_i$}
2.      If Not-Visited = {} Then
          return Fail
3.   Pop first node S from Not-Visited and push it in Visited
4.   Expand node S Then
     4.1. If $S_j$ is final state Then
               The Solution is ($S_j$, $S_{j-1}$, .., $S_i$)
          .    return SUCCESS
     4.2.Else
             Push reversed All Successors of $S_j$ in Not-Visited
5.      repeat from step 2
```



Uninformed Search - DFS



Not-Visited	Visited
{A}	{}
{B,C}	{A}
{D, E, C}	{A, B}
{F, E, C}	{A, B, D}
{I, E, C}	{A, B, D, F}
{E, C}	{A, B, D, F, I}
{G, C}	{A, B, D, F, I, E}

Traversed Path:

{A, B, D, F, I, E, G}

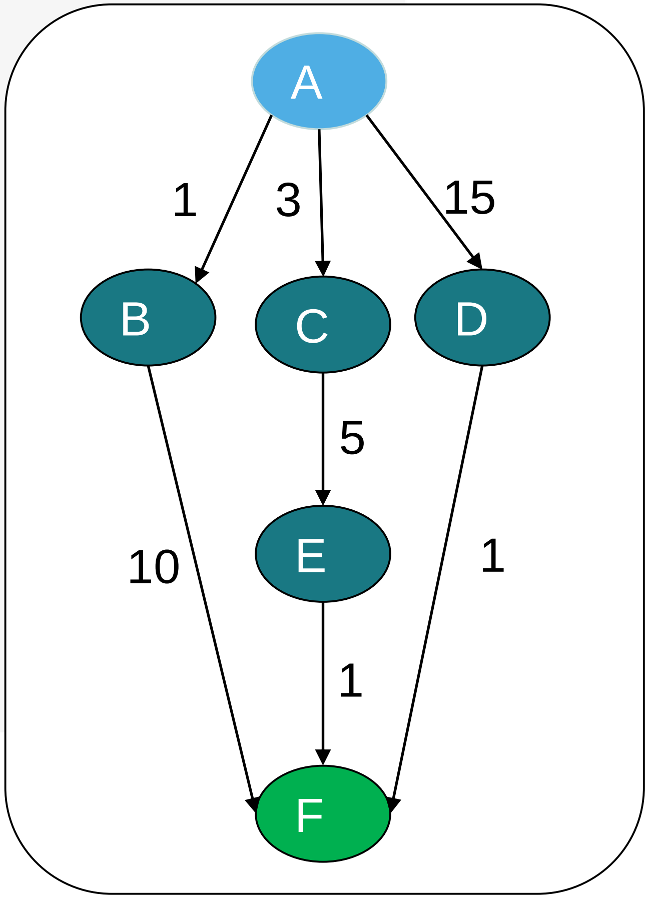
Uninformed Search - UCS

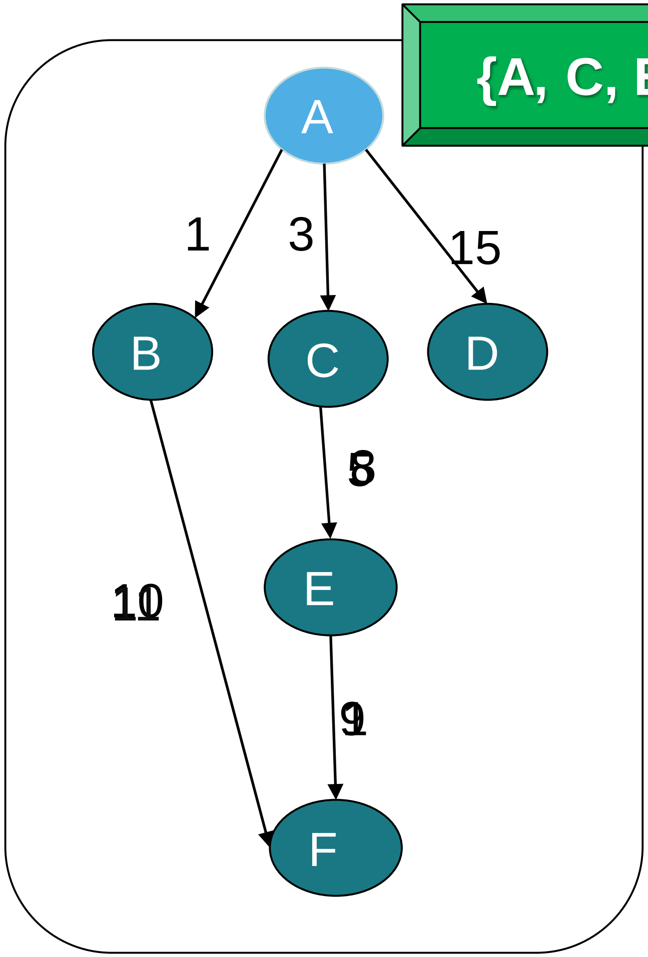
- Similar to Breath-First.
- Pursues the path with less accumulated cost.

Algorithm UCS:

```
Insert the root node into the list
While the list is not empty:
    Remove the lowest cost element from the list
    If costs are same, alphabetically smaller path is chosen Then:
        If the path is ending in the goal state, print the path and Return Success
    Else
        Insert all the children of the removed element, with the cumulative costs.
End While
Return Fail
```

Uninformed Search - UCS





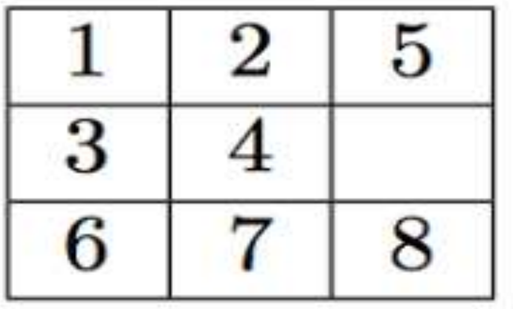
Traversed Path:

{A, C, E, F} with cost 9

Example – Puzzle 8 Game

- Given an initial state of the board, the combinatorial search problem is to find a sequence of moves that transitions this state to the goal state
- The blank space is going to be represented with the number 0.
- The blank space may be swapped with a component in one of the four directions {'Up', 'Down', 'Left', 'Right'}, one move at a time

Initial state =

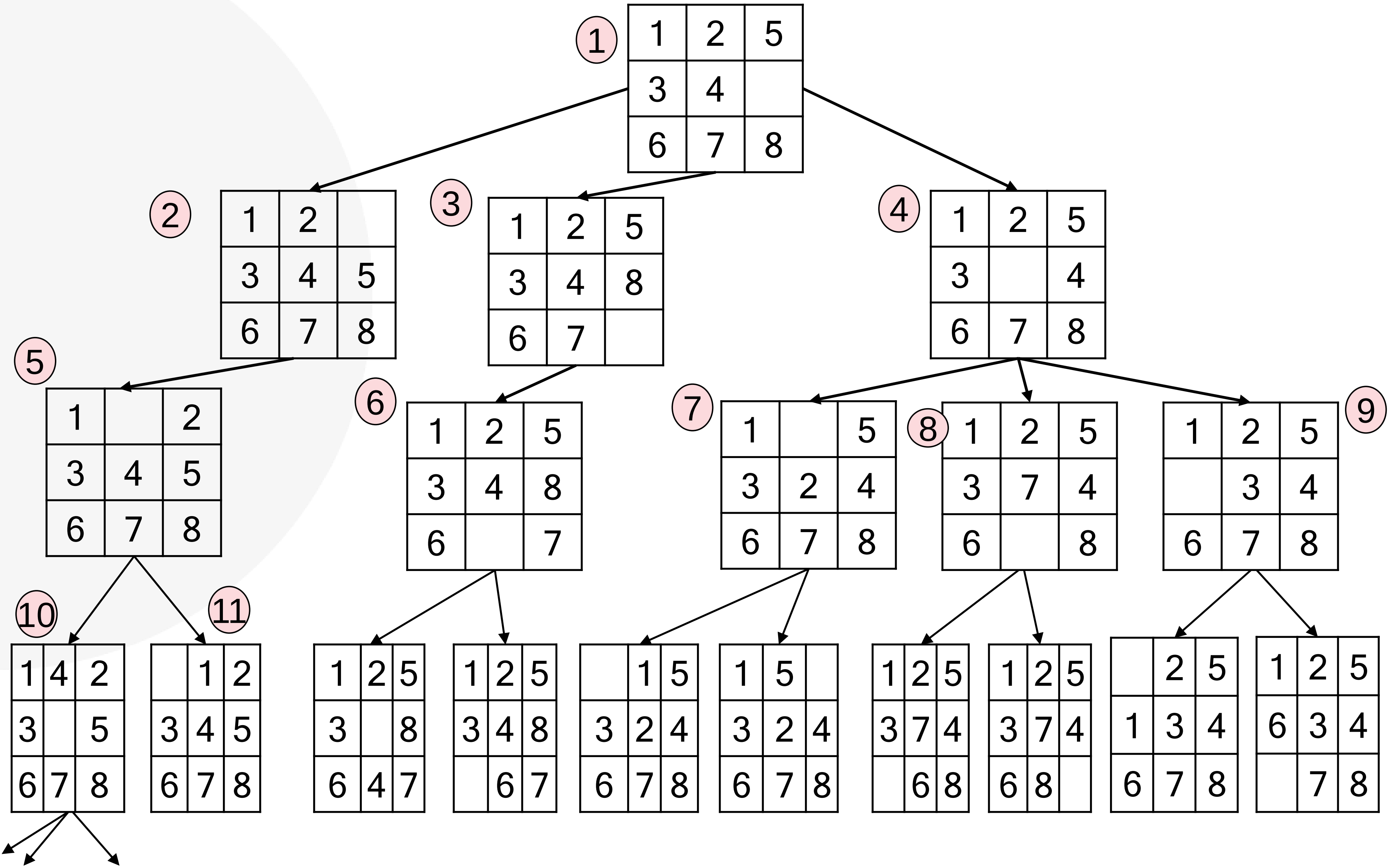


1	2	5
3	4	
6	7	8

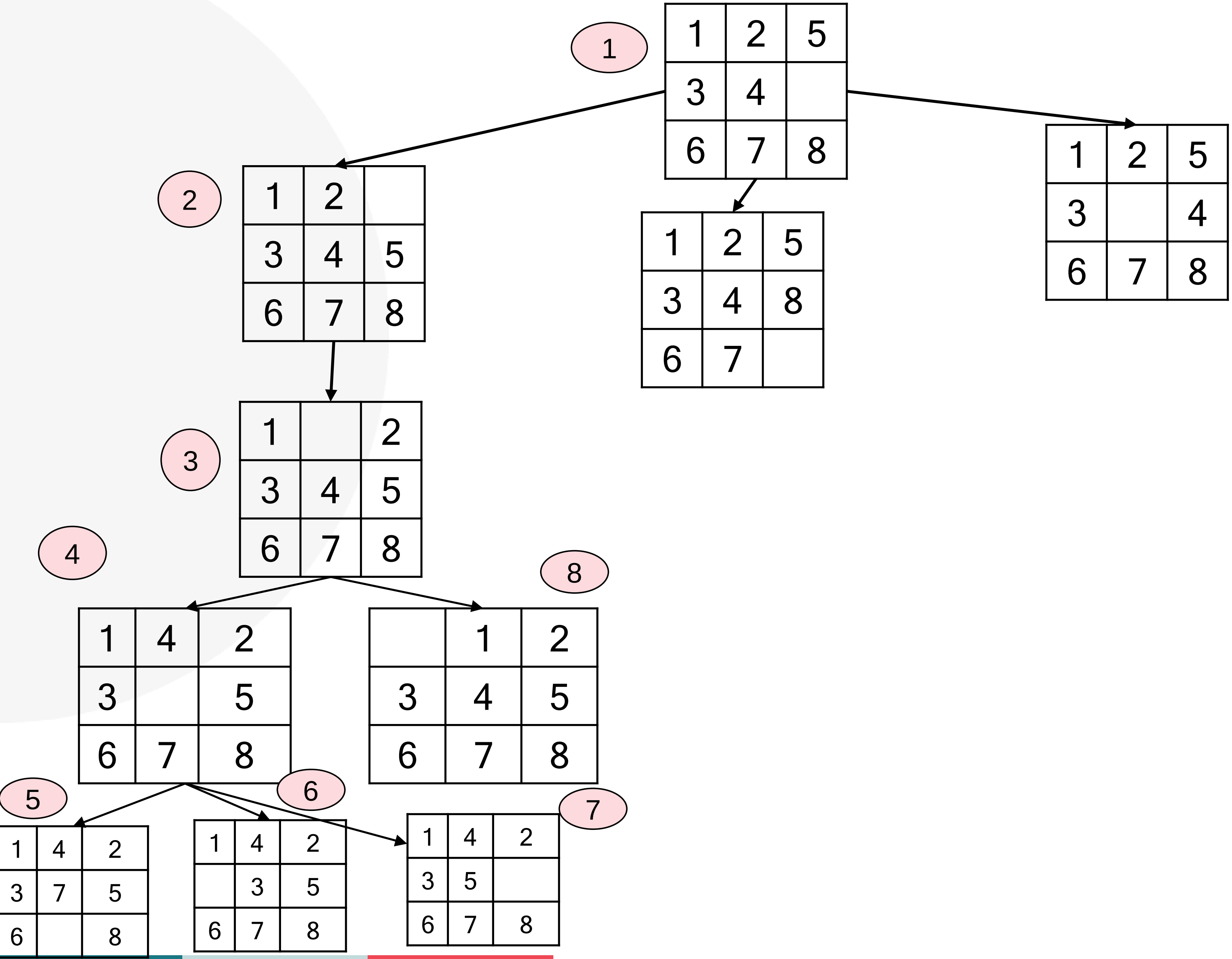
Goal State =

1	2	5
3	4	
6	7	8

8 Puzzle Game – Using BFS



8 Puzzle Game – Using DFS



1

1	2	5
3	4	
6	7	8

2

1	2	
3	4	5
6	7	8

1	2	5
3	4	8
6	7	

1	2	5
3		4
6	7	8

3

1		2
3	4	5
6	7	8

4

1	4	2
3		5
6	7	8

8

	1	2
3	4	5
6	7	8

5

1	4	2
3	7	5
6		8

6

1	4	2
	3	5
6	7	8

7

1	4	2
3	5	
6	7	8

Hands On – Puzzle Board

Implement BFS to solve 8 puzzle game:

Start state is: [1,2,5,3,4,#,6,7,8]

While # is denoted for the space.

Print the path for the goal and number of iterations you needed to reach that goal.

Solution

Milestone 2

Uninformed Search milestone deadline: 23 march 2019.
It will be published on course-sites.

General instructions:

- Regarding your AI-Package:
- Add a new folder named ‘UninformedSearch’.
- Add only one new ‘.PY’ file for writing your code.

Questions?