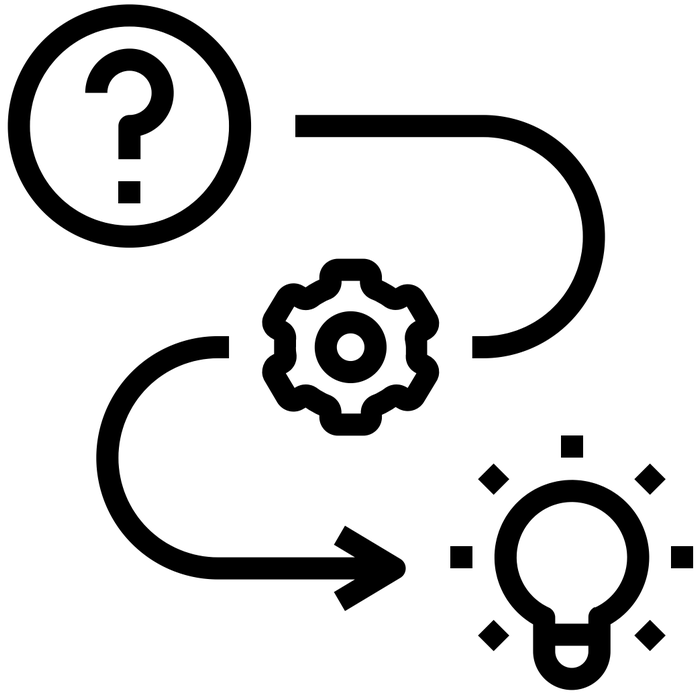
ARTIFICIAL INTELLIGENCE

BACS2003|BACS3074|BMCS2003

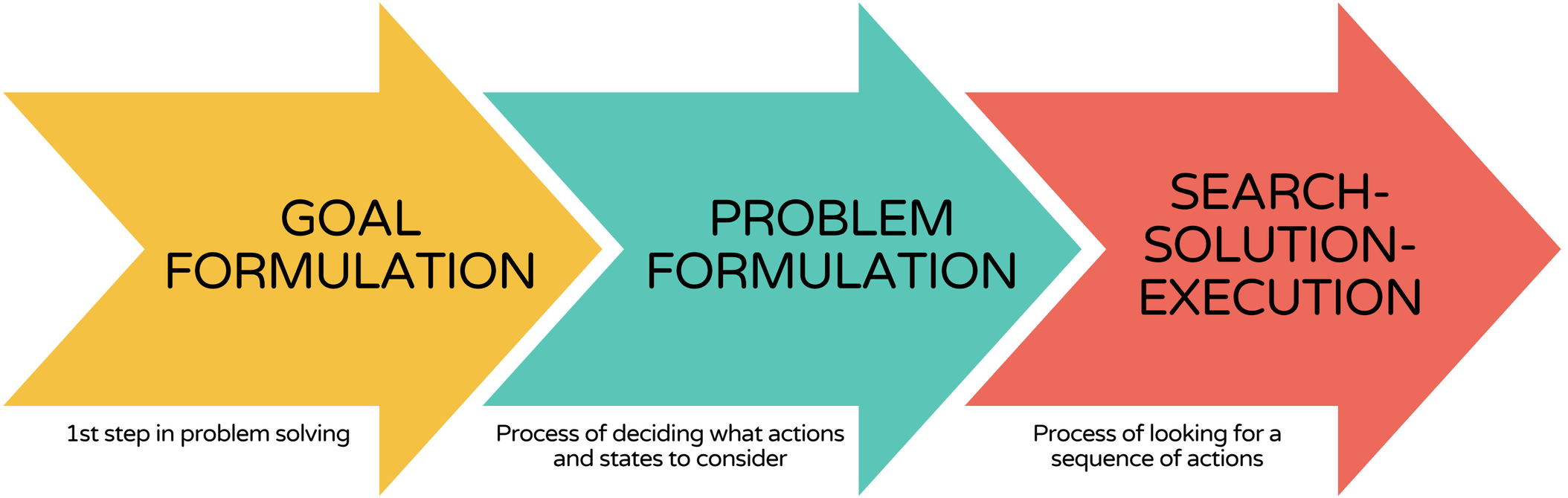
CHAPTER 2 PROBLEM DEFINITION AND PROBLEM SOLVING

OUTCOMES

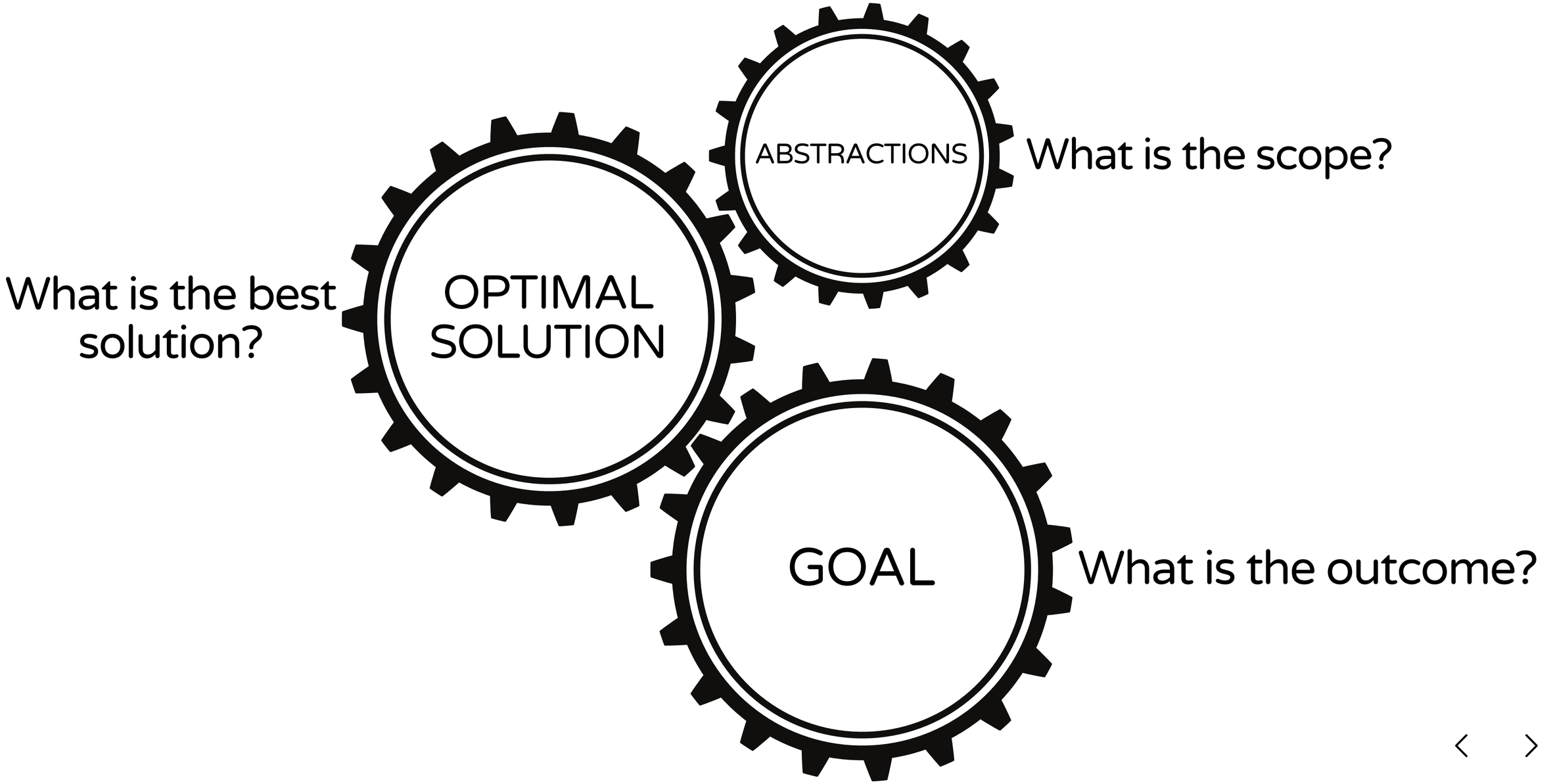
1. **Goal formulation**
2. **Problem formulation**



PROBLEM-SOLVING CONCEPT

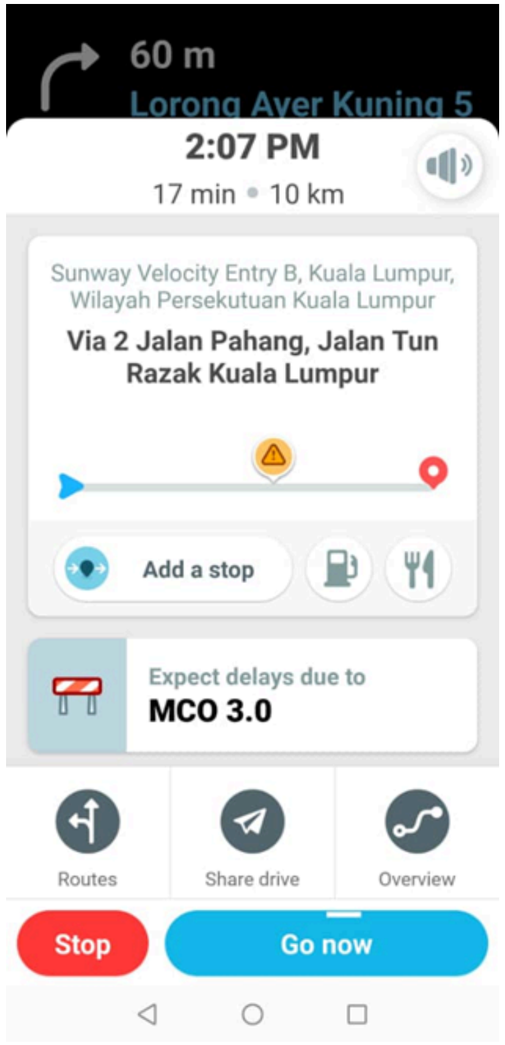


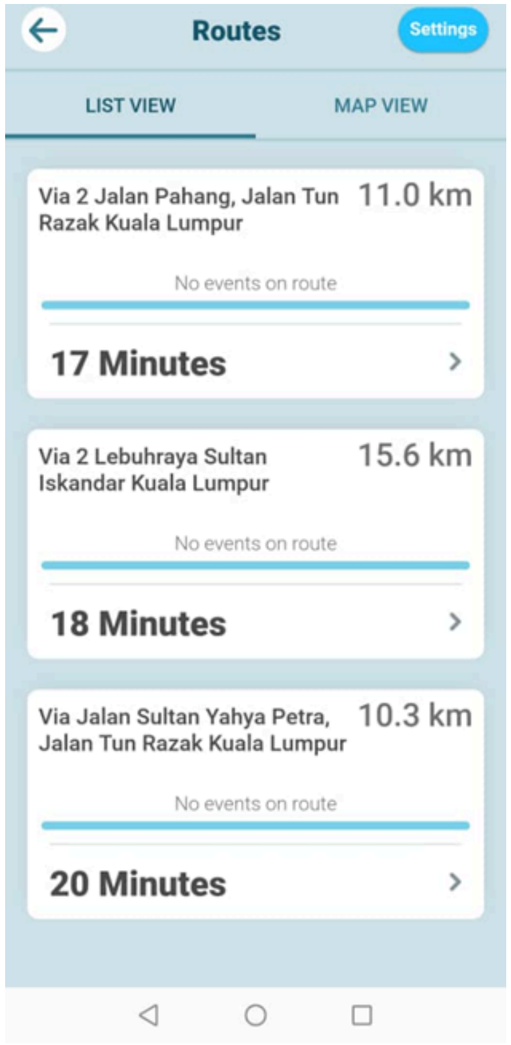
STEP 1: GOAL FORMULATION

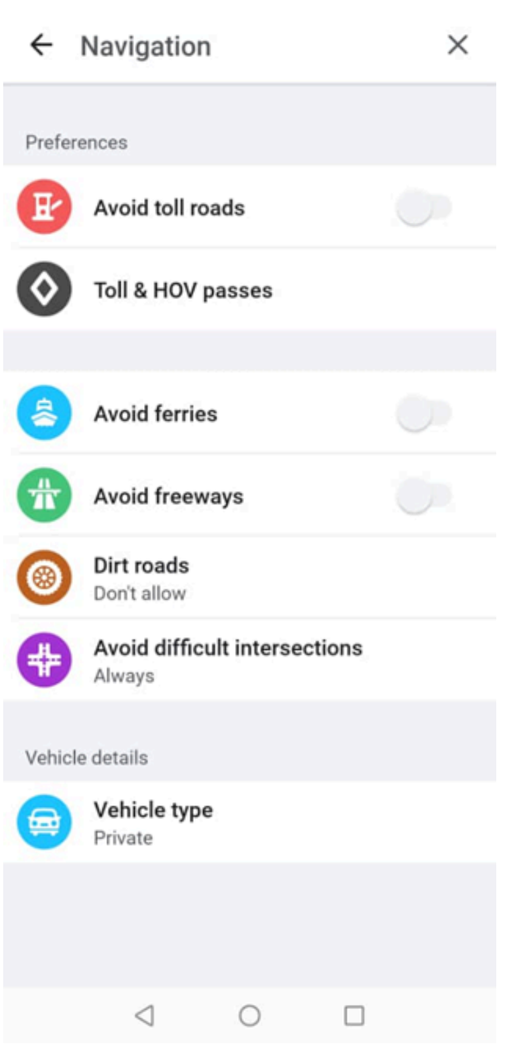


EXAMPLE 1

PATH FINDING







EXAMPLE 2

PATH FINDING

Destination

15KM

13KM

8KM

3KM

Departure

6KM

EXAMPLE 2

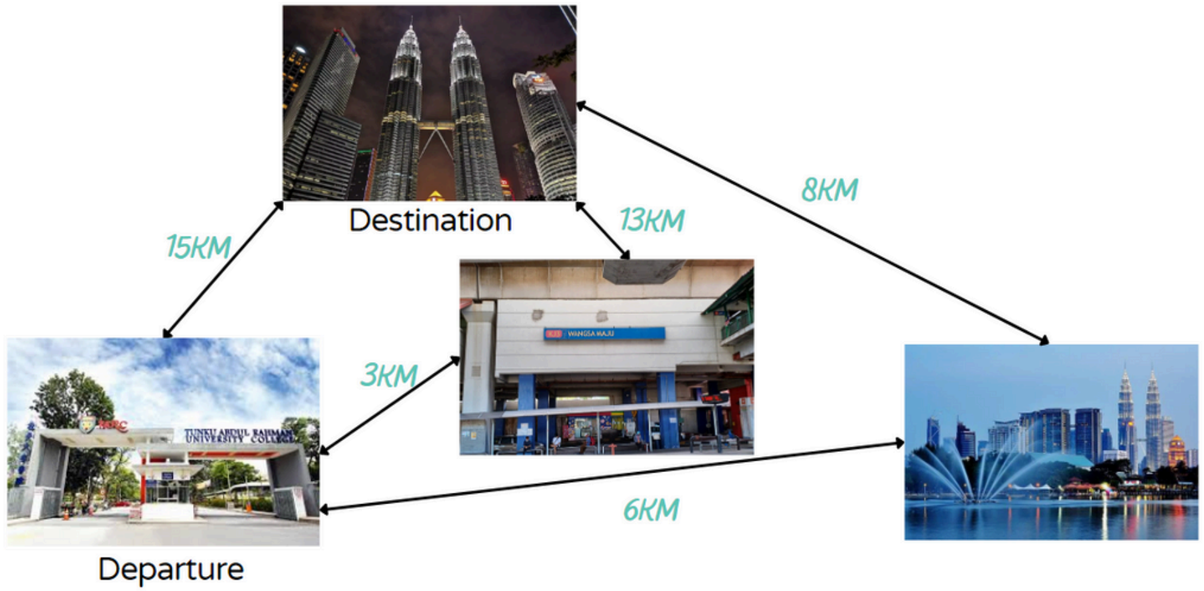
PATH FINDING

STEP 1: FORMULATING GOAL

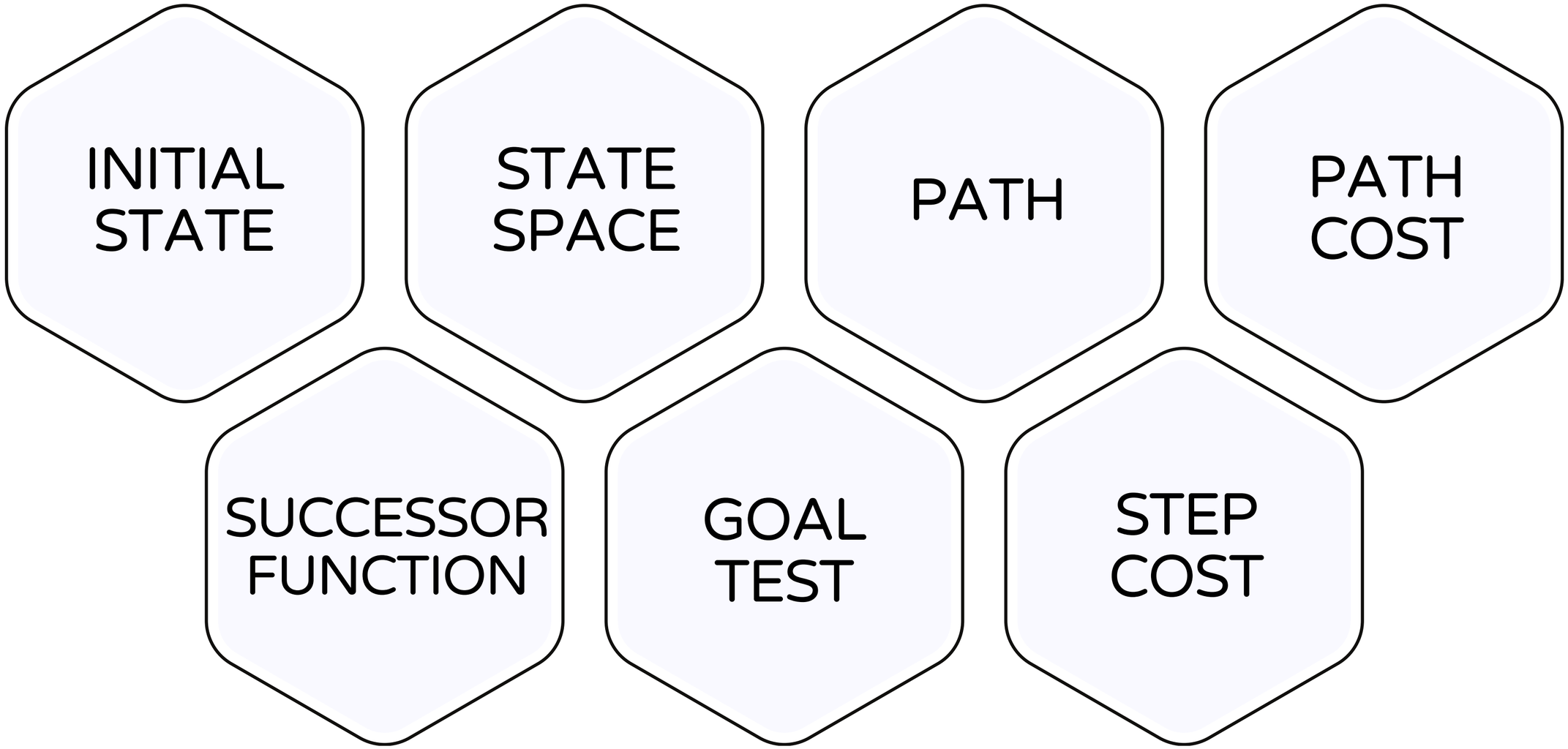
Goal

Optimal Solution

Abstraction



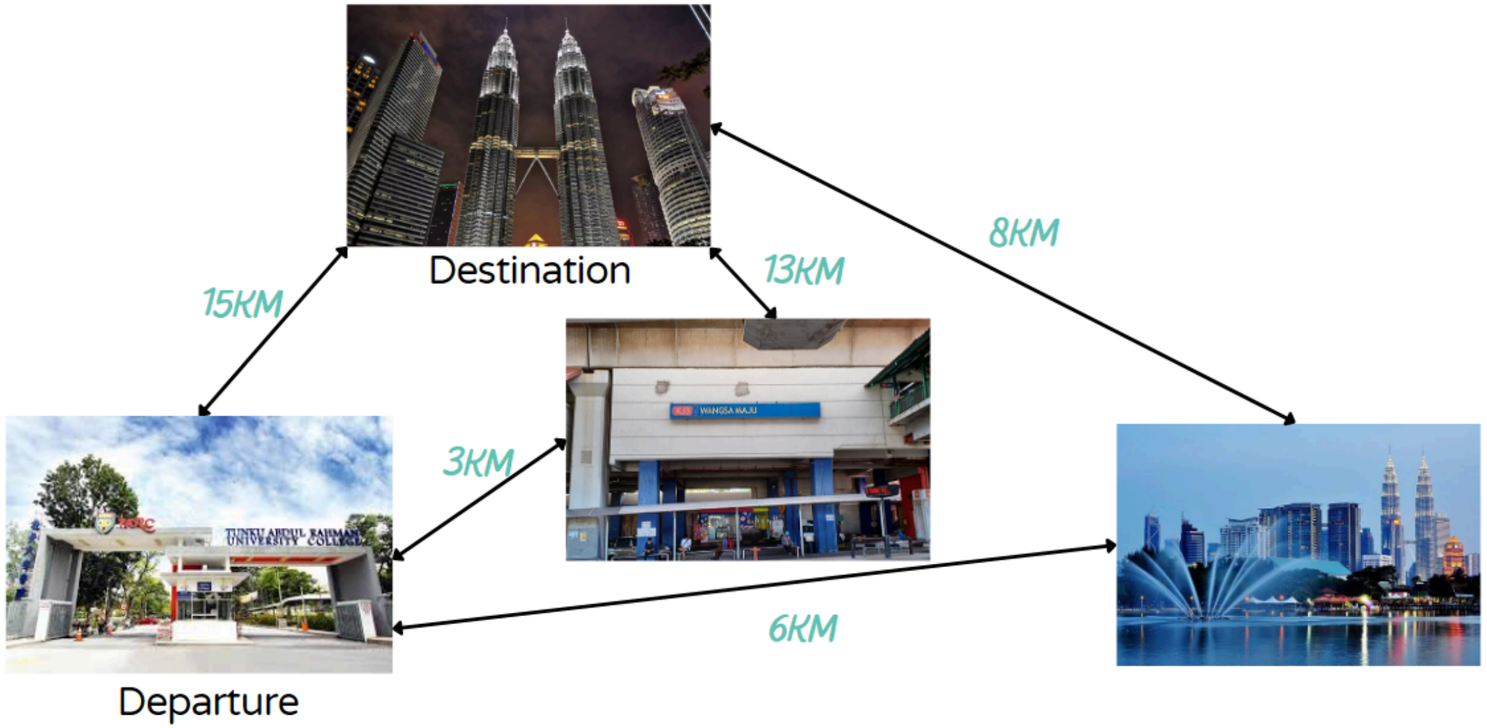
STEP 2: FORMULATING PROBLEM



EXAMPLE

PATH FINDING

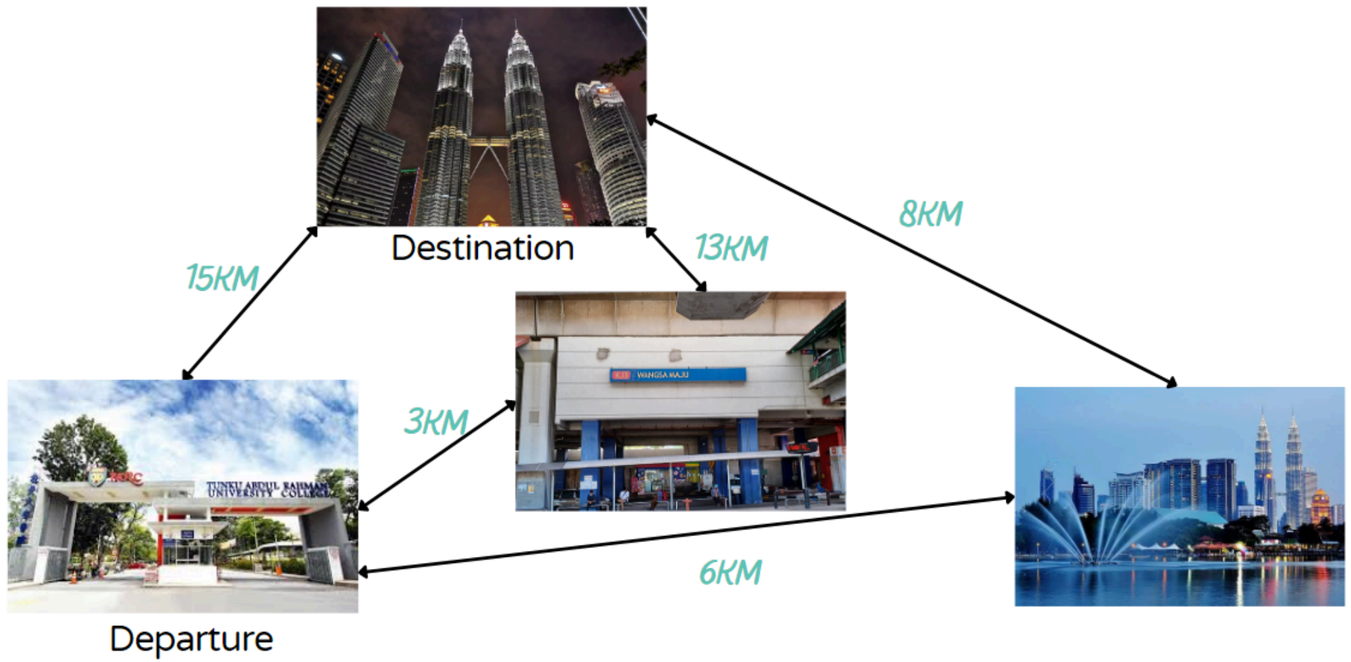
STEP 2: FORMULATING PROBLEM



Initial state (the state that the agent starts in)
E.g.
Go(_ , In(TARUMT), 0)

EXAMPLE

STEP 2: FORMULATING PROBLEM



PATH FINDING

Successor function (the possible Actions available to the agent, that will change the state)

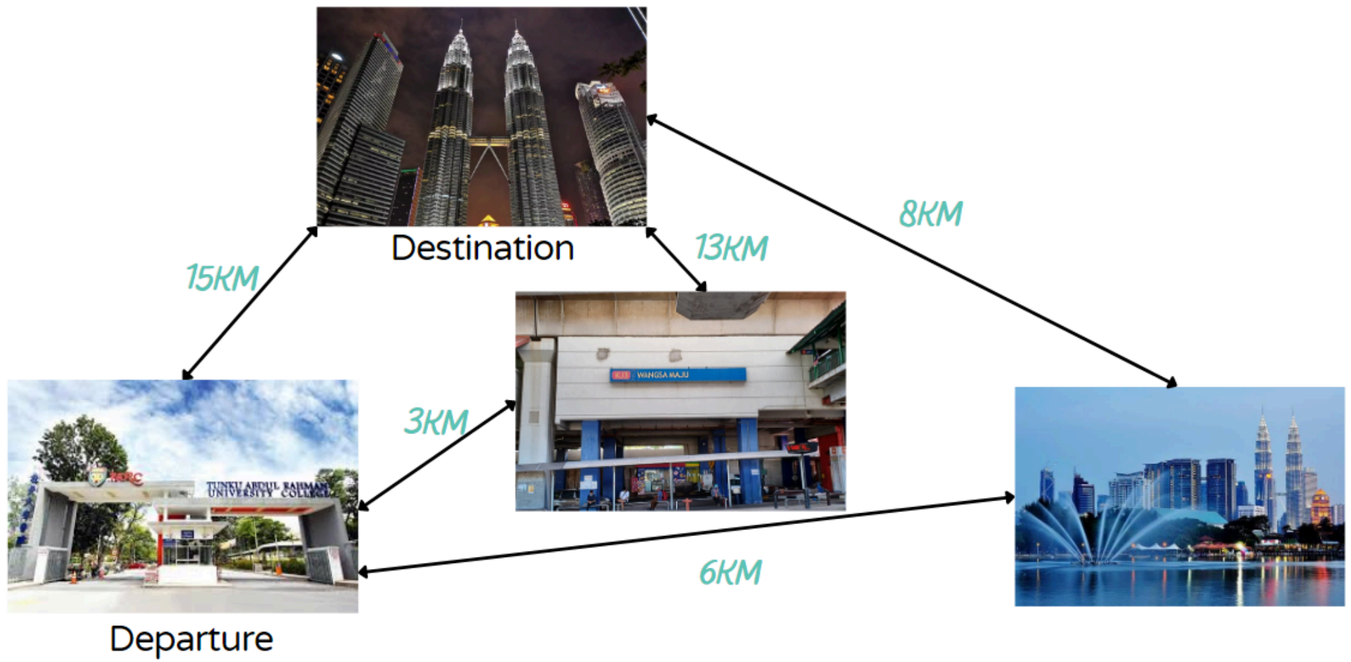
For example, a function can be given as: Go(To(child), In(Parent), State_Cost)

E.g.:

- Go(To(Wangsa Maju, In(Tarumt), 3)
- Go(To(Titiwangsa, In(Tarumt), 6)
- Go(To(KLCC), In(Wangsa Maju), 13)
- Go(To(Tarumt), In(Wangsa Maju), 3)

EXAMPLE

STEP 2: FORMULATING PROBLEM

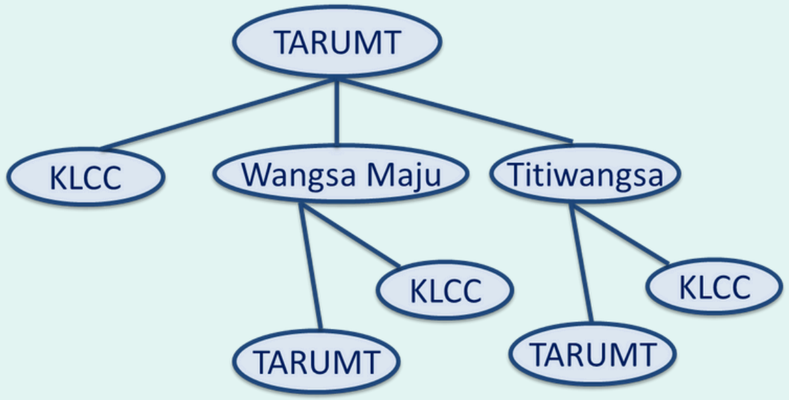


PATH FINDING

State space (the set of all states reachable from the initial state)

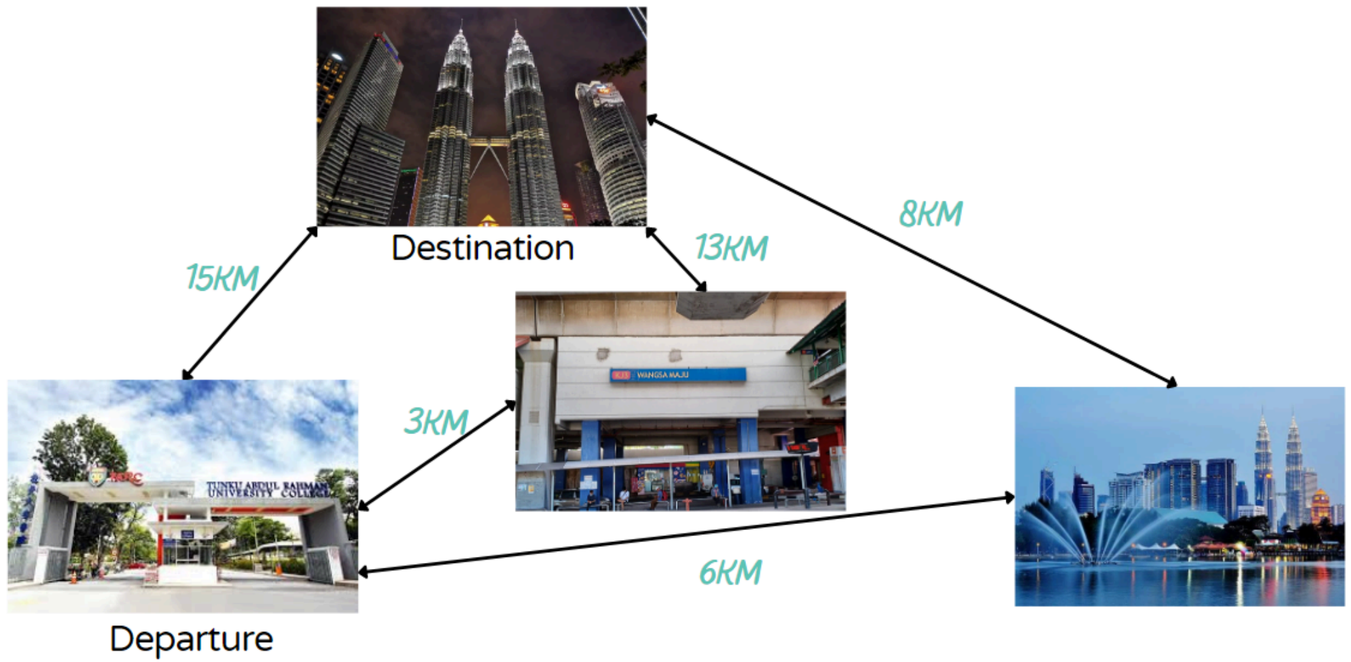
Usually represented by a graph or a tree

E.g.



EXAMPLE

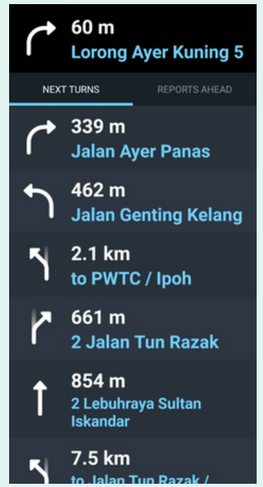
STEP 2: FORMULATING PROBLEM



PATH FINDING

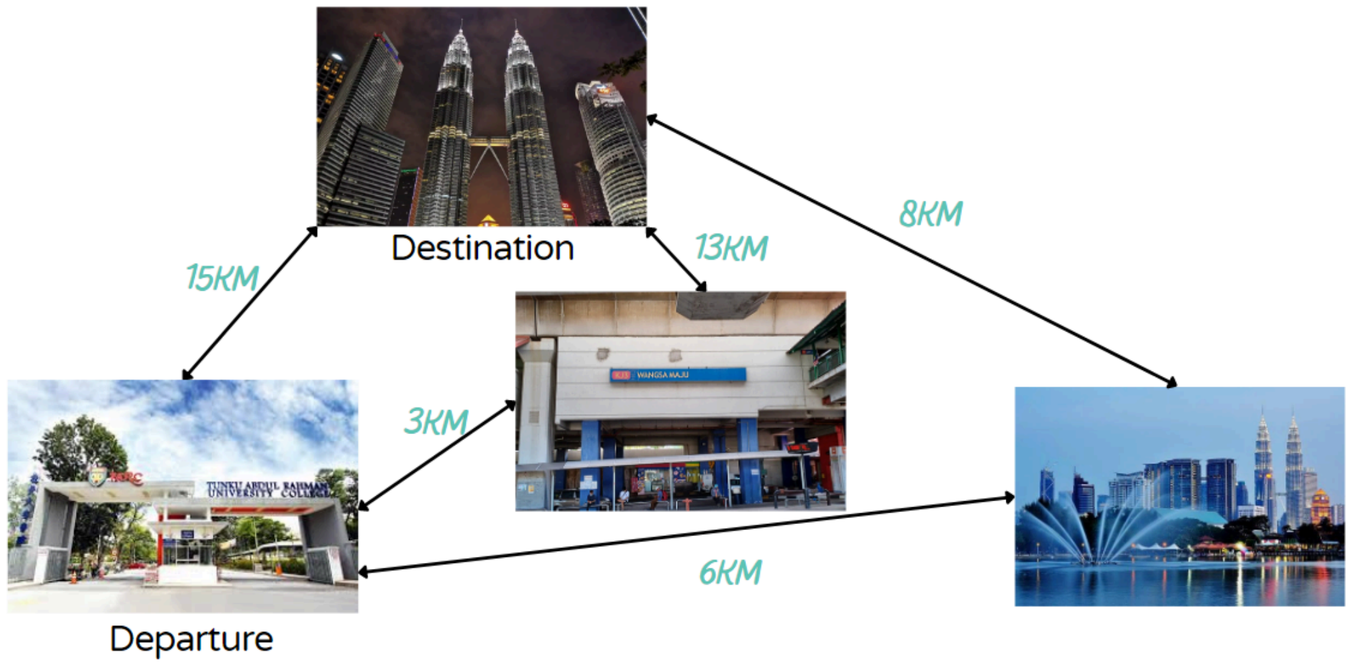
Path (Sequence of states connected by a sequence of actions)

E.G. TARUMT --> WM ---> KLCC



EXAMPLE

STEP 2: FORMULATING PROBLEM



PATH FINDING

Goal Test (To determine whether a given state is a goal state)

Sample of Goal Test Algorithm

E.G.

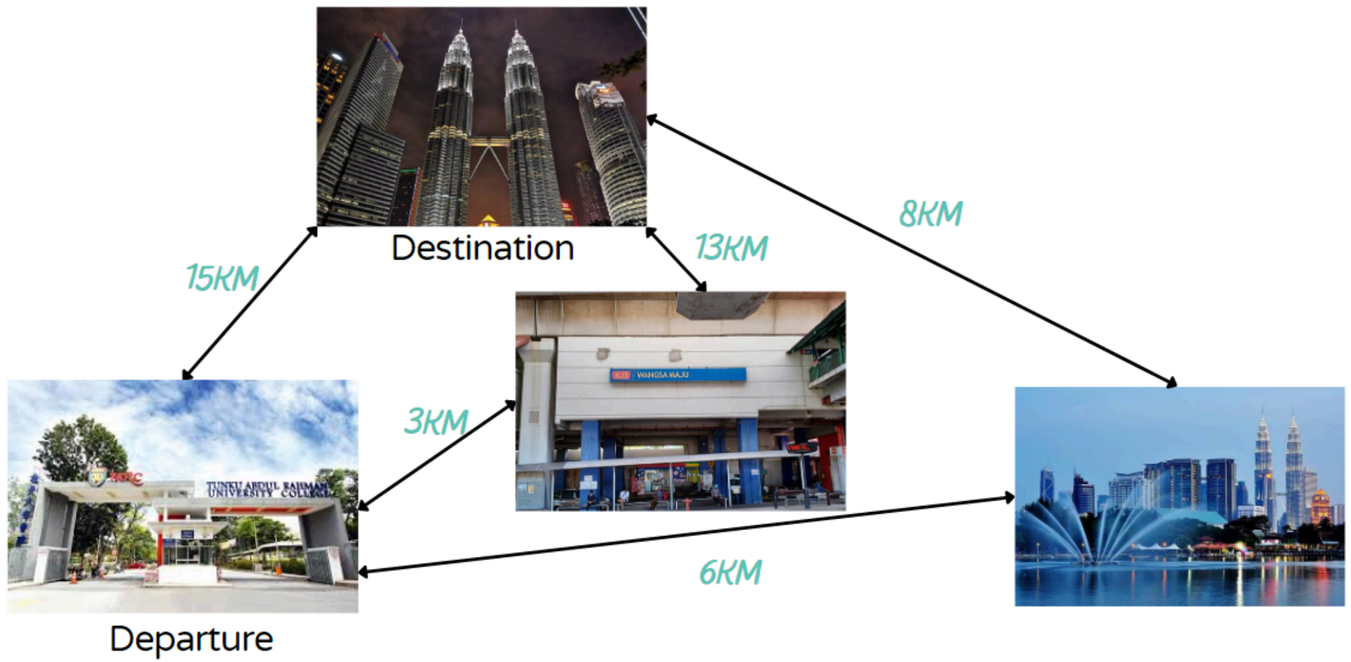
```
If
   Go( _ , In(KLCC), _)
Then
   show pathcost

 return true
```

#current state == goal state

EXAMPLE

STEP 2: FORMULATING PROBLEM



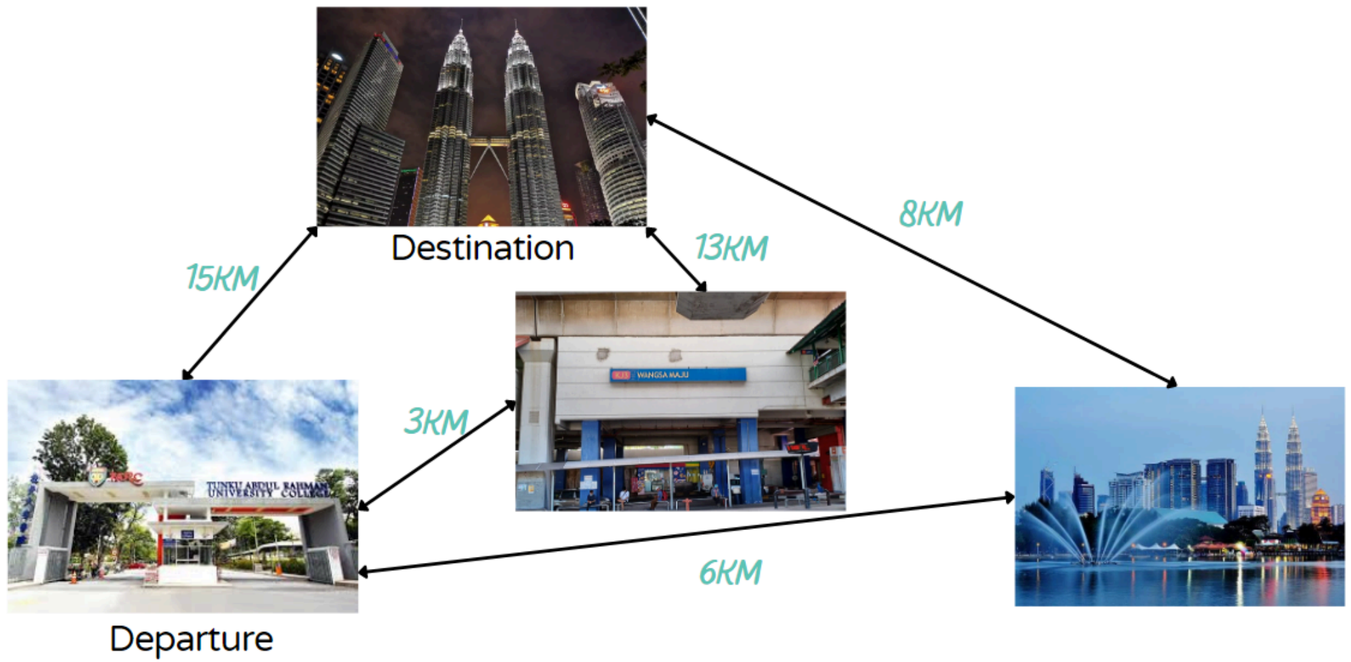
PATH FINDING

Step cost (route distance, from one state to another state)

E.G.
the step cost from TARUMT to WangsaMaju = 3km

EXAMPLE

STEP 2: FORMULATING PROBLEM

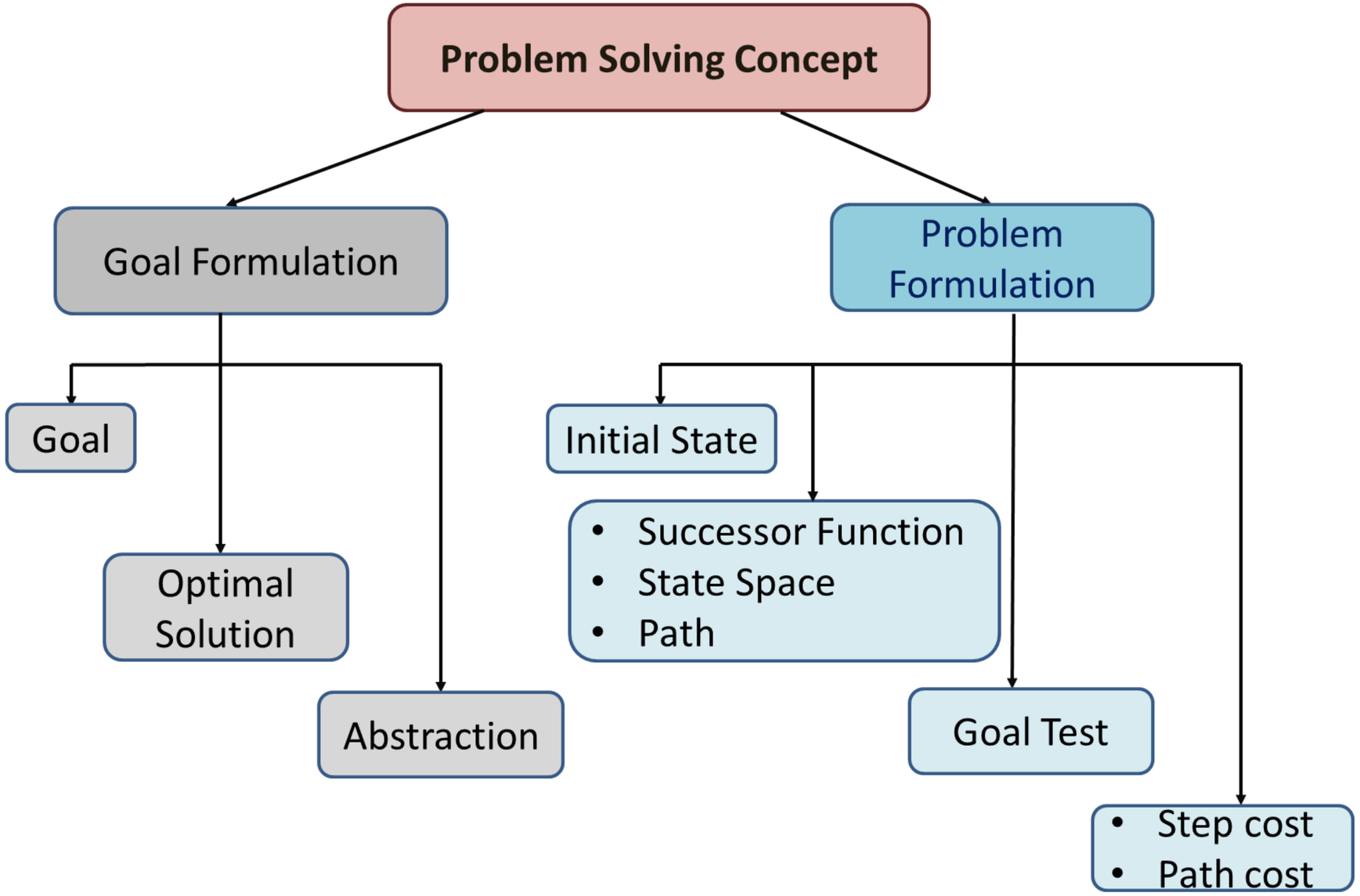


PATH FINDING

Path cost (Sum of the costs of the individual actions along the path)

E.G.
From TARUMT to KLCC :16 = 3+13

SUMMARY

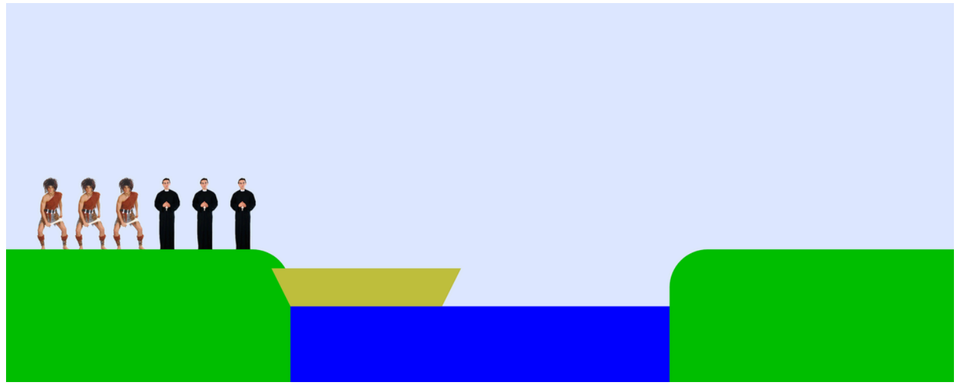

Problem Solving Concept
Goal Formulation
Problem Formulation
Goal
Optimal Solution
Abstraction
Initial State
Successor Function
State Space
Path
Goal Test
Step cost
Path cost

QUESTION

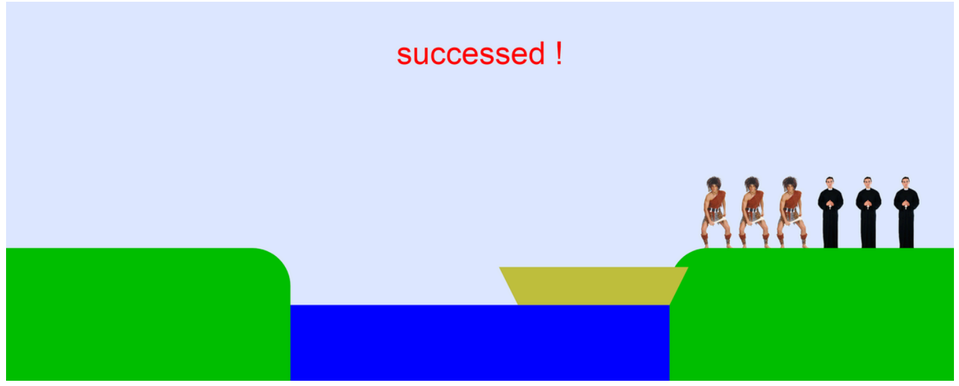
MISSIONARIES & CANNIBALS

FORMULATE GOAL & PROBLEM

INITIAL STATE



GOAL



Rule:

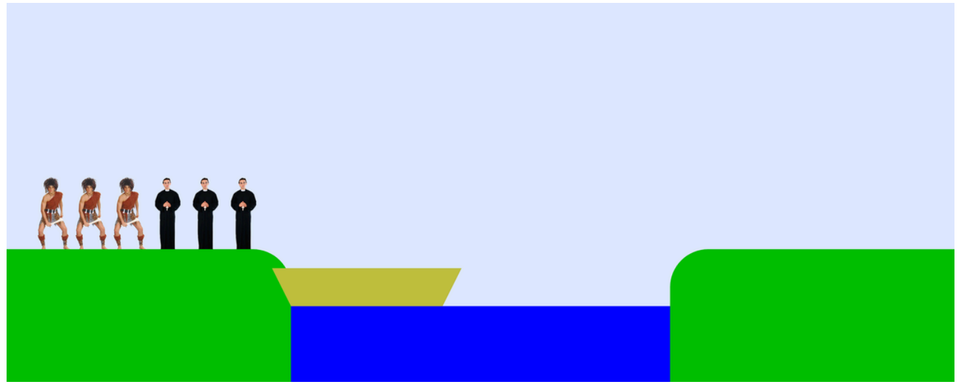
1. Missionaries >= cannibals
2. No one is allowed to swim
3. Max 2 people/trip

QUESTION

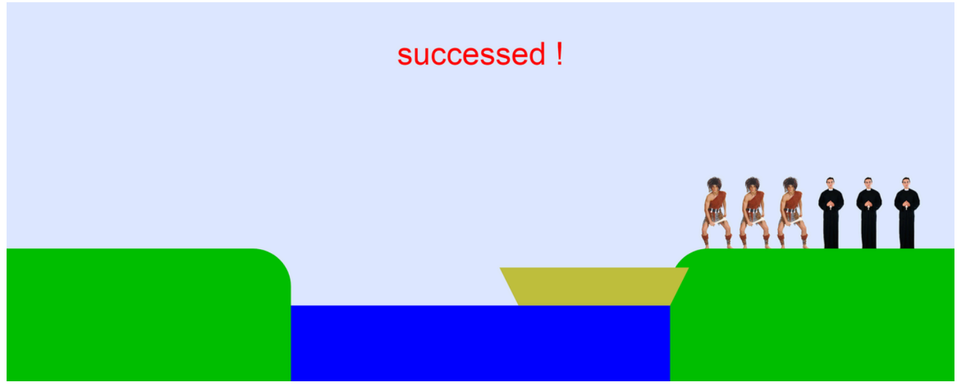
MISSIONARIES & CANNIBALS

FORMULATE GOAL

INITIAL STATE



GOAL



Goal:

Optimal solution:

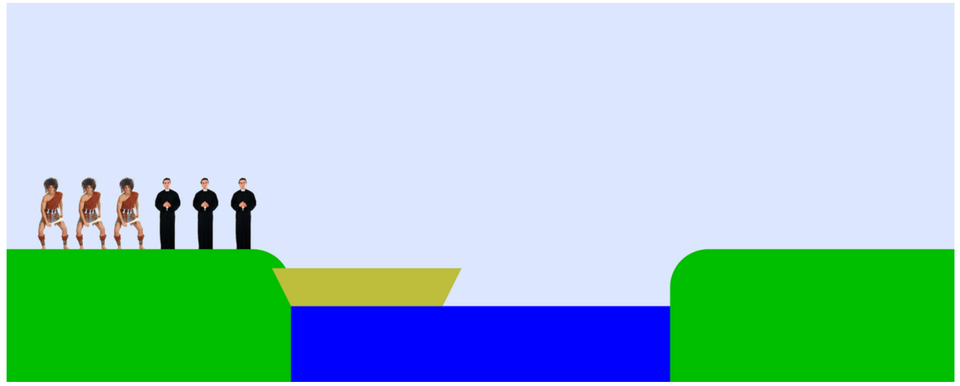
Abstraction:

QUESTION

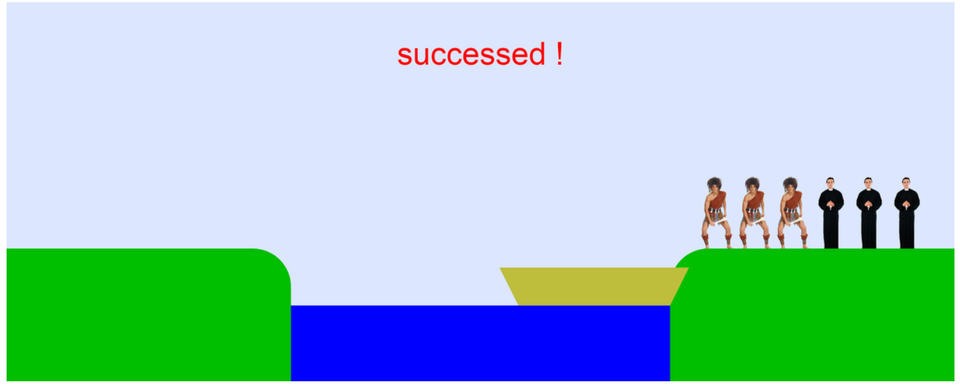
MISSIONARIES & CANNIBALS

FORMULATE PROBLEM

INITIAL STATE



GOAL



Initial state:

Successor function:

State space:

Path:

Goal test:

Step cost:

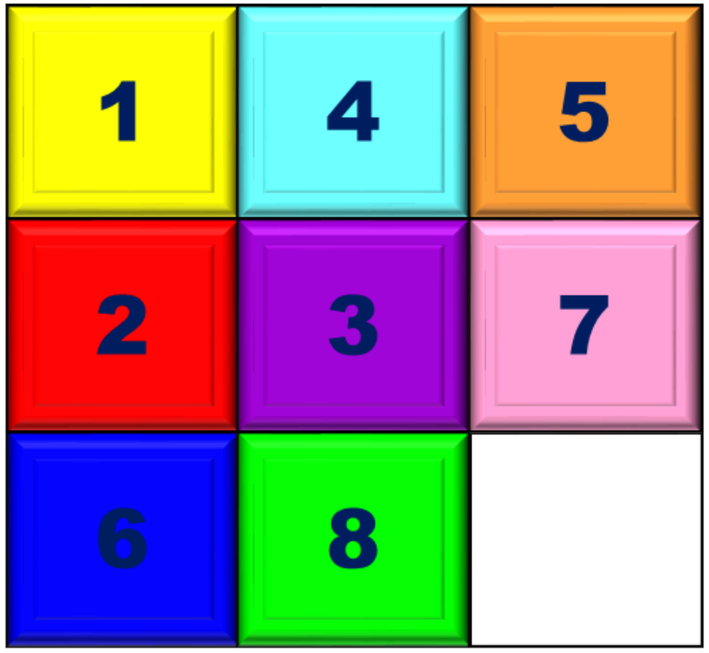
Path cost:

QUESTION

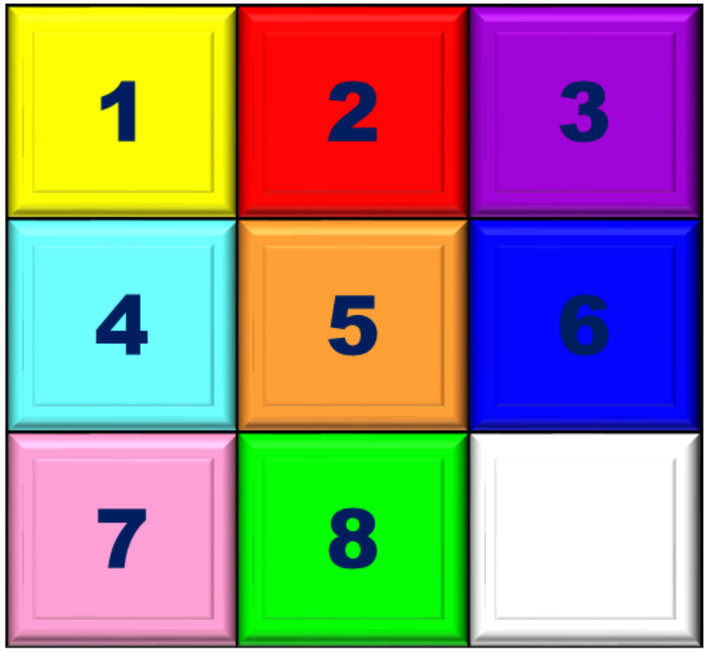
THE 8 SLIDING PUZZLE

FORMULATE GOAL & PROBLEM

INITIAL STATE



GOAL



Rule:

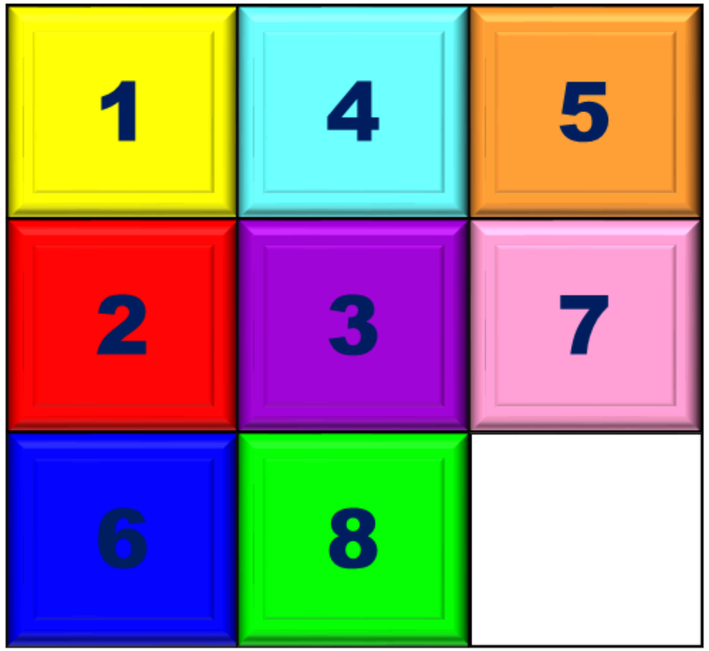
1. Only slide is allowed
2. Not allowed to slide over the boundary

QUESTION

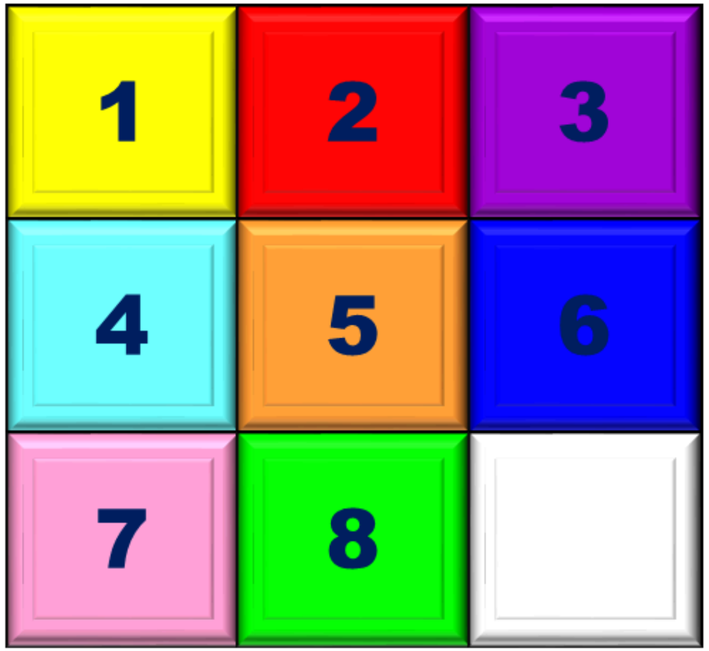
THE 8-PUZZLE

FORMULATE GOAL & PROBLEM

INITIAL STATE



GOAL



Goal:

Optimal solution:

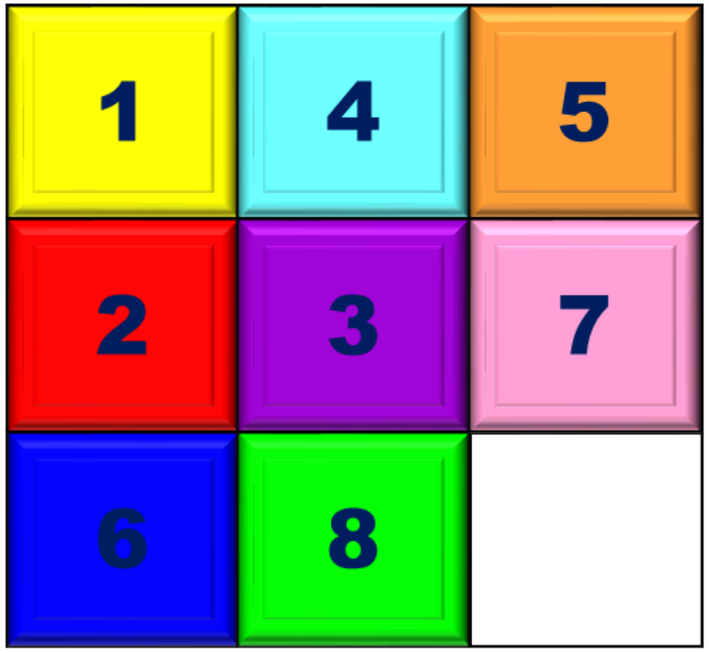
Abstraction:

QUESTION

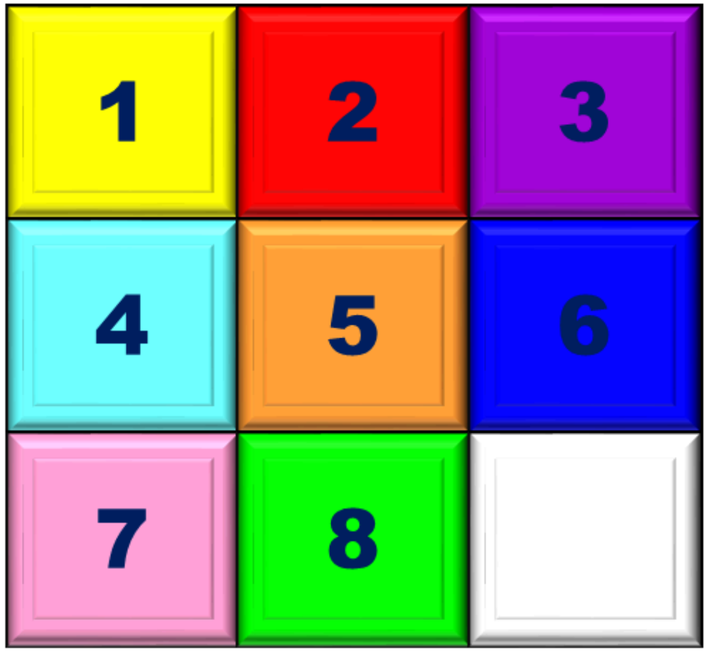
THE 8-PUZZLE

FORMULATE GOAL & PROBLEM

INITIAL STATE



GOAL



Initial state:
Successor function:
State space:
Path:
Goal test:
Step cost:
Path cost:

SEARCH-SOLUTION-EXECUTION

Search

- the process of looking for the best sequence of path

Solution

- A search algorithm takes a problem as input and returns a solution in the form of an action sequence

Execution

- Once a solution is found, the actions it recommends can be carried out.

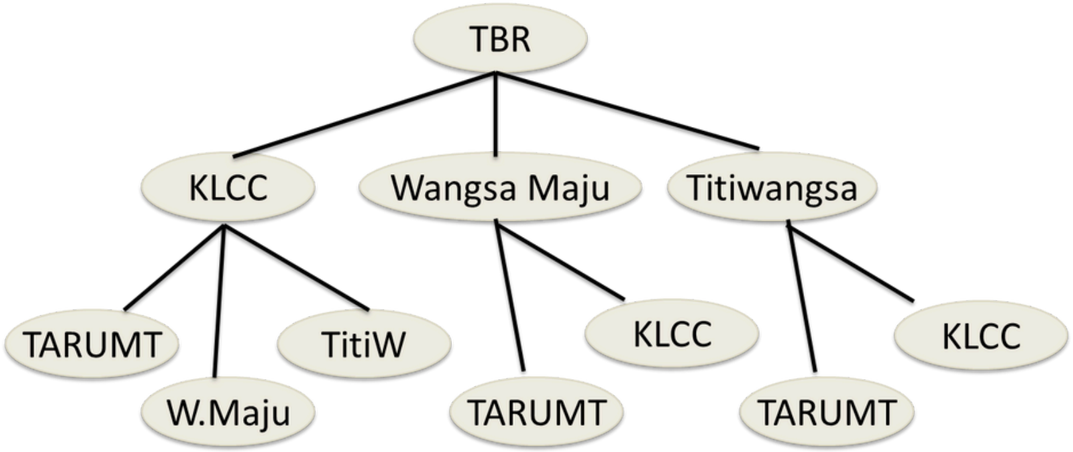
SEARCHING FOR SOLUTIONS

- Search tree
- Search graph
- Expanding/generating states
- Search strategy

SEARCH TREE VS SEACH GRAPH

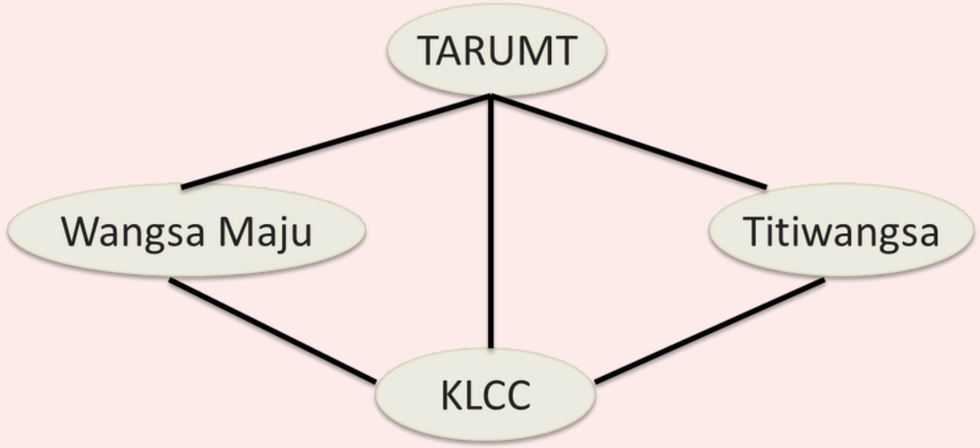
SEARCH TREE*

Generated by the initial state and the successor function that together define state space.

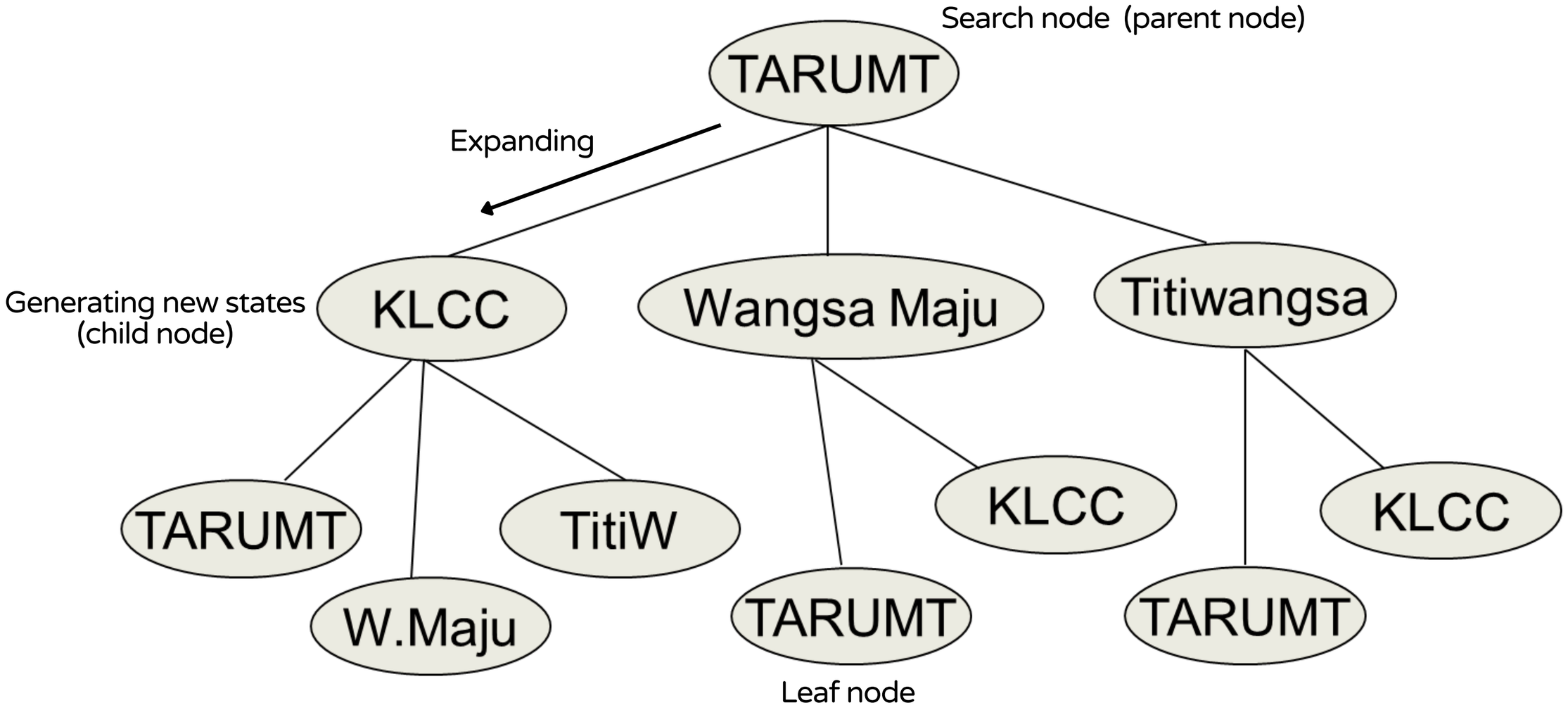


SEARCH GRAPH

The same state can be reached from multiple paths.



THEORIES OF SEARCH TREE

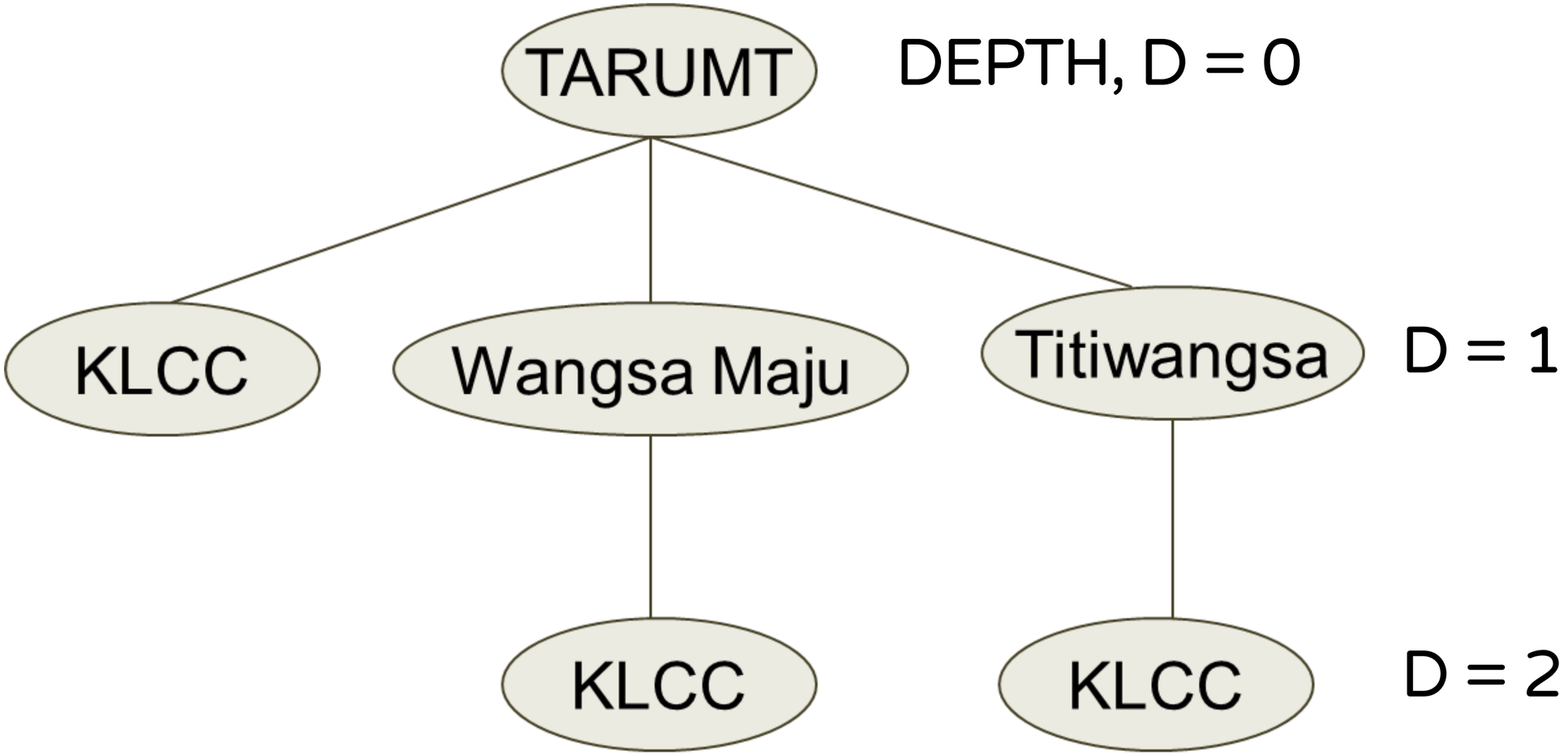


Fringe:Use to store the nodes on the frontier of our traversal's discovery process (the next nodes it is waiting to look at)

AVOIDING REPEATED STATES

- Wasting time
- Can cause a solvable problem become unsolvable if the algorithm does not detect them
- How to solve?

SEARCH TREE WITHOUT REPEATED STATE



MEASURING PROBLEM-SOLVING PERFORMANCE

Completeness

- Is the algorithm guaranteed to find a solution when there is one?

Optimality

- Does the strategy find the optimal solution?

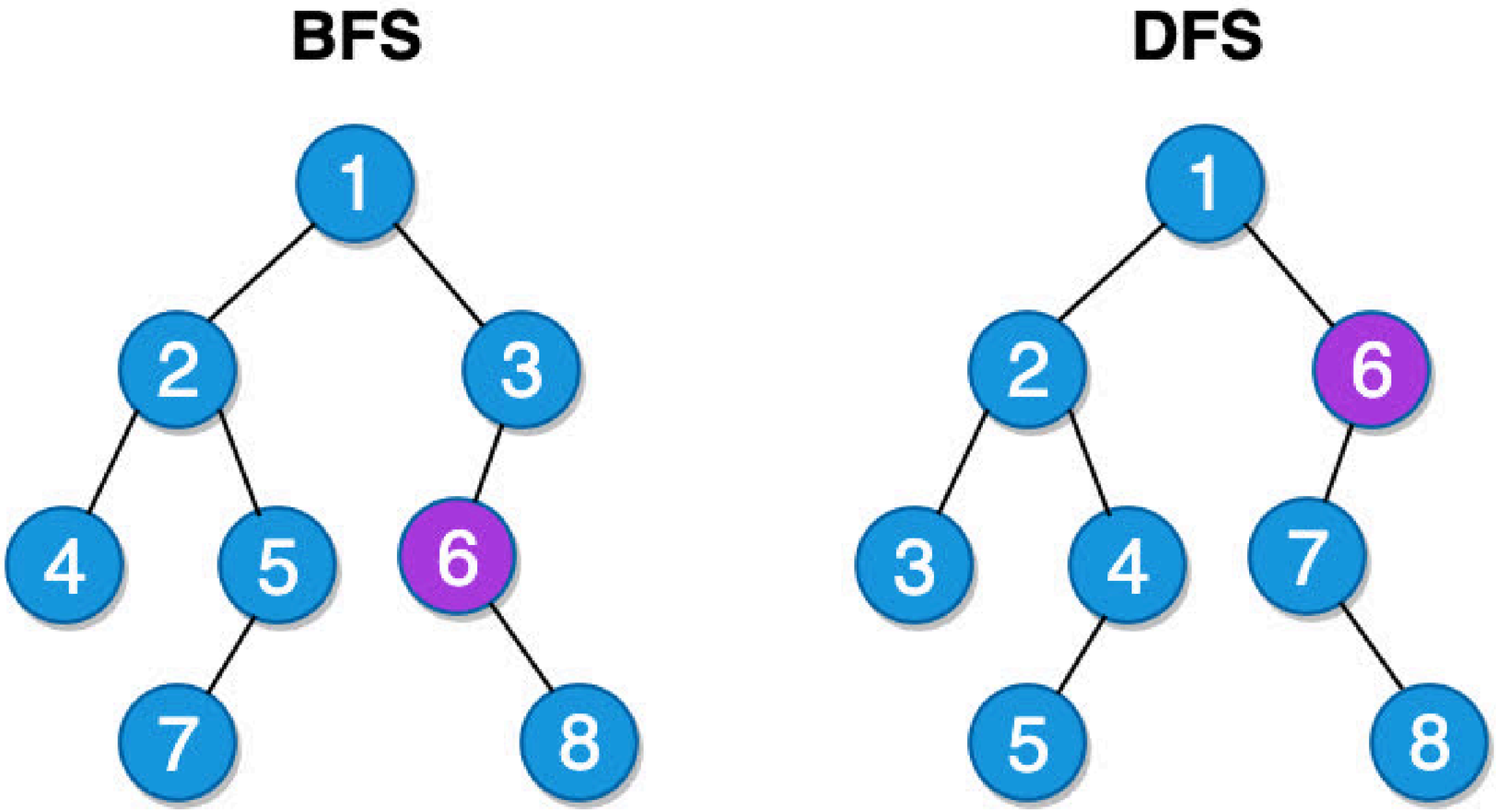
Time complexity

- How long does it take to find a solution?

Space Complexity

- How much memory is needed to perform the search?

EXAMPLE OF SEARCHING SOLUTION



THE END

NEXT LECTURE

Uninform Search