# ACI Group 1 – Assignment 1 Theory

## Environment Definition:

Performance Measures:

1. Safety of Group
2. Finding a route adjacent to the river
3. Minimize path length
4. Correct categorization of safe, hazardous, and impassable terrain

Environment:

1. Group
2. Kangaroo Island
3. Fire
4. River
5. Passable Terrain

Actuators:

1. Motors on the Rescue Agent
2. Signal for various terrain categories
3. Speakers for interacting with the group

Sensors:

1. MWIR Infrared cameras
2. Proximity Sensor
3. Speedometer
4. GPS
5. Temperature
6. Camera

## Algorithm 1 - Uninformed Search

Based on the available information about the problem, the Search Algorithm is first categorized into Uninformed Search Algorithm, which generates the search tree without any domain-specific knowledge or additional information about the states beyond the information provided in the problem definitions. This class-purpose search algorithm, which is also known as Blind Search or Brute-Force Search, operates in a brute-force way and examines each root node of the search tree until it achieves the goal state.

### Requirements of Uninformed Search:

Though Uninformed Search does not require excessive details about the problem, it does have a few search requirements, like:

**State Description:** States that are reachable from the initial state by any sequence of actions as well as details like where the search begins, ends, transition model, etc.

**A valid set of Operations:** That helps find the most optimal solution with the lowest cost.

**Initial State:** Information about the stage where the agent first starts the search.

**Goal State Description:** Details like the limit of the search tree, the sequence of nodes from start to goal, etc. should be considered.

### Components Included in Uninformed Search:

In Uninformed Search Algorithms, each of the six search strategies covers the following components, which becomes helpful at a different stage of problem-solving. These components are:

**Problem Graph:** Contains the start and goal nodes.

**Strategy:** Describes the way the graph will get to the goal.

**Fringe:** Includes the data structure used to store all the possible states that can be achieved from the current state.

**Tree:** It is the result of traversing from the start node to the goal node.

**Solution Plan:** This is the sequence of nodes from the start to the goal node.

### Breadth-First Search

The most common search strategy, breadth-first search, starts from the root node, explores the neighboring nodes first and moves towards the next level. Implemented using the First-in First-out (FIFO) queue data structure, this is an example of a general-graph search algorithm that provides the shortest path to the solution.

The algorithm uses a queue to remember to get the next vertex to start a search as well as when a dead end occurs in any iteration. Without this queue, it won’t be able to identify, and mark visited vertices and will process them again, which can then become a non-terminating process.

## Algorithm 2 – Recursive Best First Search

Recursive best-first search is a simple recursive algorithm that improves upon heuristic search by reducing the memory requirement. RBFS uses only linear space, and it attempts to mimic the operation of standard best-first search. Its structure is like recursive depth-first search, but it doesn't continue indefinitely down the current path, the f-limit variable is used to keep track of the f-value of the best alternative path available from any ancestor of the current node. RBFS remembers the f-value of the best leaf in the forgotten subtree and can decide whether it is worth re-expanding the tree later. However, RBFS still suffers from excessive node regeneration.

## Complexity

For BFS, the time complexity is a function of the number of traversed nodes, or O(V), since in the worst case, all nodes are visited. Space complexity follows a similar function, also being O(V).

In the case of RBFS, its space complexity is represented by O(bd) where b is the branching factor (number of children at node) and d is maximum search depth. Time complexity is represented by O(b^d) with b and d representing the same values as above.