# Assignment Sheet
# EVEN Semester 2021
# B.Tech CSE/IT 6th Semester

# Artificial Intelligence Lab (15B17CI574)

---

**Instructions:**

- Students have to do a mini project apart from the Lab Assignments.
- The evaluative lab assignments must be evaluated as per the given deadline. The total weightage of all day to day work is 60 Marks.
- There will be two lab tests of 20 marks each. Absence in Lab Test-2 means Fail in the lab course.
- All students are required to attend at least 80% labs. 15 marks are reserved for attendance.
- The evaluative lab assignments must be evaluated as per the given deadline from time to time.

---

## Week-3

## Febuary 1-6, 2021

**EXERCISES:**

**1.** Write a procedure that is analogous to list referencing, but for trees. This "tree_ref" procedure will take a tree and an index, and return the part of the tree (a leaf or a subtree) at that index. For trees, indices will have to be lists of integers. Consider the tree in Figure 1, represented by this Python tuple: (((1, 2), 3), (4, (5, 6)), 7, (8, 9, 10)).To select the element 9 out of it, we'd normally need to do something like tree[3][1]. Instead, we'd prefer to do tree_ref(tree, (3, 1)) (note that we're using zero-based indexing, as in list-ref, and that the indices come in top-down order; so an index of (3, 1) means you should take the fourth branch of the main tree, and then the second branch of that subtree). As another example, the element 6 could be selected by tree_ref(tree, (1, 1, 1)). Note that it's okay for the result to be a subtree, rather than a leaf. So tree_ref(tree, (0,)) should return ((1, 2), 3).

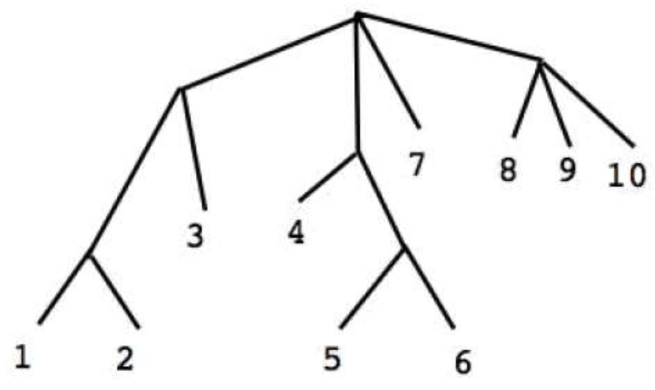


Figure 1: Example Tree

**2.** Write a program that reduces nested expressions made of sums and products into a **single sum of products**. For example, it turns the expression (2 * (x + 1) * (y + 3)) into ((2 * x * y) + (2 * x * 3) + (2 * 1 * y) + (2 * 3)). You could choose to simplify further, such as to (2 * x * y) + (6 * x) + (2 * y) + 6, but it is not necessary.

**Take 5 different inputs and analyze the time and space complexity of all the algorithms in Q3, Q4, Q5, Q6.**

**3.** Write a python program to implement Breadth First Search traversal.

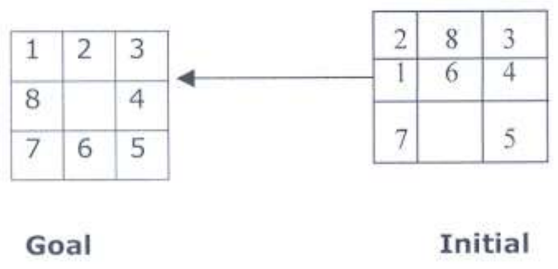
**4.** Write a program to implement of Depth First Search.

**5.** Write a program to implement Uniform Cost Search.

**6.** Write a program to implement Iterative deepening Search.

**7. 8 Puzzle Problem**

The 8 puzzle consists of eight numbered, movable tiles set in a 3x3 frame. One cell of the frame is always empty thus making it possible to move an adjacent numbered tile into the empty cell. Such a puzzle is illustrated in following diagram.



The program is to change the initial configuration into the goal configuration. A solution to the problem is an appropriate sequence of moves, such as "move tiles 5 to the right, move tile 7 to the left,move tile 6 to the down, etc". Solve this problem using any of the uninformed strategy.

**8. Water-Jug Problem:**

You are given a m litre jug and a n litre jug . Both the jugs are initially empty. The jugs don't have markings to allow measuring smaller quantities. You have to use the jugs to measure d litres of water where d is less than n. Provide a solution using BFS or DFS.

(State1, State2) corresponds to a state where State1 refers to amount of water in Jug1 and State2 refers to amount of water in Jug2. Determine the path from initial state (State1_initial, State2_initial ) to final state (State1_final, State2_final), where

(State1_initial, State2_initial ) is (0, 0) which indicates both Jugs are initially empty and (State1_final, State2_final)indicates a state which could be (0, d) or (d, 0).