1. Introduction

This project aims to develop a tree-search algorithm using Python. Initially, the general concept and structure of a search tree will be outlined, along with an explanation of the various search types implemented in the code. The focus will then shift to the Python implementation, including details on code structure and data representation. Finally, the algorithm will be tested on some example problems to evaluate its performance and compare the different search strategies.

2. What is a Tree Search algorithm?

2.1. Subtitle2

2.1.1. Subtitle3

2.1.1.1. Subtitle4

2.1.1.1.1. Subtitle5

2.1.1.1.2. Subtitle5-1

If we consider problem in which the possible states can be represented as nodes of a tree, and the actions can be represented by edges between nodes, we can define a search strategy that builds a tree structure in which the first node is the starting status, and the children are the states that can be reached with a particular action. This can be continued to build a tree-like structure. The goal of the algorithm is, starting from the problem definition, build a tree that represents the problem and find a path that connects the starting node to the goal node through a list of actions.
There exists multiple search algorithms and strategies, but they can be subdivided in one of these two classes:

- Informed search algorithm
- Uninformed search algorithm

To explain this concept, let's consider an example in which a search algorithm is applied to find a path between two points in a map. In this problem each intersection is represented as a state and the edges are the action to move from an intersection to another. We know intuitively that starting from a certain point, to find a possible solution is better to "go" in the general direction of our goal.
An uninformed search algorithm doesn't know if a particular path is leading to a solution state or not. This means that these algorithms can start searching in a direction that will never lead to a solution.
However, some search algorithms have a way to tell if a path or direction is good or not.This algorithm are called Informed search algorithms. In the previous example, an informed search algorithm can use the distance between the current position and the goal to evaluate if it's moving in a good direction, toward the target, or not. Using this information the search is more goal-driven and usually more efficient.

Figure 1: uhuhuh

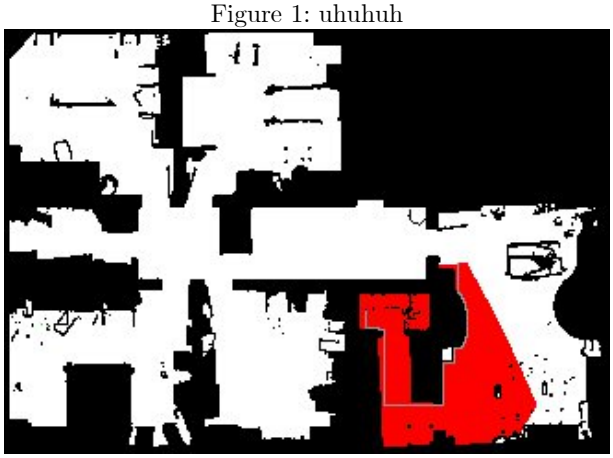

Figure 2: uhuhuhx2

pla0	Pla pla pla, some more text, some more more more more text
pla3