**Homework 1**

**CSE 5120 (Section 02) – Introduction to Artificial Intelligence – Spring 2022**

*Submitted to*

**Department of Computer Science and Engineering**

**California State University, San Bernardino, California**

by

Corbin Ulloa (006973376)

Date: February 27, 2022

Email:

* 006973376@coyote.csusb.edu

**Report**

1. **Depth First Search**  
   DFS expands the deepest unexpanded node as it is a LIFO data structure. I used Stack in my code to implement DFS algorithm.

A screenshot of a computer screen

Description automatically generated with medium confidence

1. **Breadth First Search**  
   Your brief explanation (e.g., does BFS expand the shallowest or deepest unexpanded node? did you use Stack, Queue, or PriorityQueue in your code?) with screenshots of your code Evaluation (results from   
   autograder.py)  
   BFS expands the shallowest unexpanded node as it is a FIFO data structure. I used Queue to implement BFS algorithm.

**A screenshot of a computer screen

Description automatically generated**

1. **Uniform Cost Search**  
   Your brief explanation (e.g., does BFS expand the cheapest or closest node to the goal state? What function did you use to expand the cheapest or closest node in this algorithm and at which line?) with screenshots of your code Evaluation (results from autograder.py)

Expands the cheapest node to the goal state. I used the push() function at line 177 to achieve this. A screenshot of a computer screen

Description automatically generated

1. **A\* Search**  
   Your brief explanation (e.g., does A\* use g(n) or h(n)? Where in the code are using retrieving the cost of an unexpanded node to plan and which function did you implement/use to get g(n), h(n), f(n) etc?) with screenshots of your code Evaluation (results from autograder.py)

A\* uses h(n) which is the distance from the starting node to the goal node. Lines 225 – 227 of code is where I use a push() function to retrieve cost of an unexpanded node to plan. A screenshot of a computer

Description automatically generated with medium confidence