CS 451 ASSIGNMENT 1 REPORT

I implemented and ran 3 algorithms on Java for the CS 451 Assignment 1. These algorithms were breadth firs search, depth-first search, and uniform cost search. The assignment was about a HR department, which wants to hire employees for different departments. The goal was to hire required employees to the required departments with use of these algorithms.

Assumptions

- The number of candidates are limited to 10
- The required numbers of employees for the departments are set.
- There are 3 departments.

Efficiency of Algorithms

Breadth First Search:

Time complexity: O(b^d)

Depth First Search:

Time complexity: O(b^m)

Uniform Cost Search:

Time complexity: O(b^⌈C/ǫ⌉) , where C is the optimal cost of solution and ǫ is the minimal step cost

Comparison of Searches

Breadth First Search:

- Does not guarantee an optimal solution. In our case, it couldn't find the optimal solution.
- It basically doesn't increase the depth before it searches the lower nodes.

Solution For Breath First Search

After the Java code run, it prints the solution and the hired candidates. The search is able to find a feasible solution. You can see the printed solution below;

```
Feasible solution found!
Hired candidates are:
7 6 5 3 2 1
Hired candidates are: 320
```

Depth First Search:

- Does not guarantee an optimal solution. In our case, it couldn't find the optimal solution.
- It basically expands the deepest unexpanded node.
- My implementation for depth first search has a recursive way.

Solution For Depth First Search:

After the Java code run, it prints the solution and the hired candidates. The search is able to find a feasible solution. You can see the printed solution below;

```
Feasible solution found!
Hired candidates are:
7 5 4 3 2 1
Hired candidates are: 345
```

Uniform Cost Search:

- Guarantees an optimal solution.
- In my opinion, it is very similar to Dijkstra's algorithm
- Algorithm basically starts by expanding the root, then expanding the node with the lowest cost from the root, the search continues in this manner for all nodes.

Solution For Uniform Cost:

After the Java code run, it prints the optimal solution and the hired candidates. You can see the printed solution below;

```
Feasible solution found!
Hired candidates are:
10 9 6 5 2 1
Hired candidates are: 220
```

Orhun Güley S004428