

COSC 310 Program – DFS Graph – v1.0

Goal: Practice graph implementation and DFS. ~~Destroy the world.~~

Description:

CONGRATULATIONS, FELLOW HUMAN! You have been selected by the Steve K Ninja Corporation for this special project! Your continued success in assisting our organization has made us very excited for this next step. In our process for world peace, we have need for a Seek and Destroy robot that you shall help us create. We promise not to use it for evil.

Implement a graph using an Adjacency matrix to guide this robot. Starting from the first node, perform a DFS using the rules described below. Your goal is to determine the path and total cost it will take to find the targets in the graph provided. Do not use any third-party libraries, and all data structures that you use must be created by you and included with your project.

Input: A list of nodes with associated costs. Lines that begin with # are comments, please ignore them. Other lines follow the format: V W1 C1 W2 C2 W3 C3 ...

- Each line represents a vertex in the graph and all of its adjacent connections (the W's).
- V is the name of the current vertex.
- V is followed by pairs of values, W1 C1, W2 C2, etc.
 - Wn represents the name of an adjacent vertex and Cn represents its corresponding cost edge cost
 - There will be up to 10 of these pairs on a line, all separated by spaces
- Vertices are named from A to Z.
- The search begins from the first V in the file
- Vertices named X and Y designate targets. Your search will stop when it reaches one of these or return a failure.

Output: The path for our robot as a list of each node visited (in the order searched) followed by the cost it takes to reach the target if it exists. (*We promise we will use this information for non-evil tasks*).

Sample output:

```
Search1: ABDFECX, (Cost: 123)
Search2: ABCDE, Fail! (Cost: 49)
Search3: ABCABCBCX, (Cost: 100)
```

Search Specifications:

You will try three different search algorithms:

Search 1 – Alphabetic DFS

- Visit the adjacent vertices in alphabetic order. (If the adjacents are A, B, and C, visit A first).

Search 2 – Lazy DFS

- There is a good chance that our robot would say no to difficult tasks, so:
- Do not visit any vertices with an edge cost > 10
- Always select the next vertex with the lowest edge cost
- After visiting 5 vertices, give up no matter what

Search 3 – Indecisive DFS

- Vertices can be visited up to 3 times each. (To make sure we didn't miss anything!)
- Always select the next vertex that you haven't visited yet if possible
- Otherwise, select the vertex with the greatest edge cost

Tips:

- Make additional input files to test your program
- Do one thing at a time. When it works, save your code.
- Where possible, try not to harm any actual humans.

Checklist

Check	Item	Points
x	Read these specs	0
	I/O	2
	Graph	2
	DFS1	2
	DFS2	2
	DFS3	2
	Documentation	2
	Post Mortem	1
	Polish / Professionalism	2
	Typos / Missing deliverables	-Y++
	Code Errors	-Z%
-	Total	15