

Assignment 4 Design Document

Overall Assignment Overview

Overall, I have a pretty decent idea of how I want to go about approaching this assignment. First of all, this will be the basic layout of my *main* function. (Pseudocode written in Python).

```
main:
    parse_command_line
    construct_graph_from_command_line
    shortest_path = recursive_search_for_shortest_path(constructed_graph)
    print_path(shortest_path)
    exit
```

Obviously, my program is going to be much more complicated than that. This is just the basic that I want my program to adhere to.

Pseudocode

I will be providing pseudocode for some of the methods that I will be writing that are not immediately obvious.

```
graph_create(int vertices, boolean undirected):
    g = new Graph
    g.vertices = vertices
    g.undirected = undirected
    g.visited = []
    g.matrix = [[]]
    return g

graph_add_edge(Graph g, int i, int j, int k):
    g.matrix[i][j] = k
    return

graph_has_edge(Graph p, int i, int j):
    return g.matrix[i][j] != 0

depth_first_search(Graph g, int vert):
    g.mark_v_as_visited
    if(current_path_is_hamiltonian and current_path_is_shorter_than_shortest_path):
        shortest_path = current_path
    for(every edge w that can be traveled to from v):
        if(w hasn't been visited):
            depth_first_search(g,w)
    g.mark_v_as_unvisited
```

```

path_create():
    Path p = new Path
    p.vertices = []
    p.length = 0
    return p

path_push_vertex(Path p, int v, graph G):
    p.length = p.lenth + G.length_from_top_of_vertices_to_v
    p.vertices.push(v)
    return

path_pop_vertex(Path p, int v, graph G):
    v = p.vertices.pop()
    p.length = p.length - G.length_from_top_of_vertice_to_v
    return

path_copy(Path dst, Path src):
    dst.vertices = copy(src.vertices)
    dst.length = src.length
    return

    Almost all the same code for Stacks will be reused from the previous
    assignment save for one method.

stack_peek(Stack s):
    t = s.pop
    s.head++
    return t

```

Post-Coding Summary

After finishing the project, I think that I can clearly say that the most difficult part of this project for me was preventing memory leaks. I could not understand why scan-build was saying that I had potential memory leaks, and I thought at first that this was some sort of glitch with scan-build as opposed to my program. However, I did eventually find why I could have potential memory leaks, and they have since been fixed.

Thoughts

I believe that my programming style is becoming much cleaner as time progresses. I feel as if my code is more comprehensible and previous labs. I think that by creating sub methods, my code has a much more enhanced sense of readability. I still feel that I need to learn a lot more about C as I still get easily confused when thinking

about pointers, references, and other similar concepts.