

Assignment 4

Design Document (Draft)

Purpose

This program will use data from an infile to construct a graph ADT, find all hamiltonian paths using depth-first search, identify the shortest hamiltonian path, and write the path and number of recursive calls to to an outfile.

Layout/Structure

The heart of the program lies in the two functions `main` and `dfs`:

```
int main( ... )
```

Description/Explanation

In this program `main` will parse command-line options, read data from infile, construct graph, and call `dfs`

Pseudocode

Parse the command-line options

- Print help message and exit if `-h` is given

- Record if undirected or verbose printing was enabled

- Read/write with stdin/stdout unless user specifies files

- Print help message and exit if invalid option is given

Read first line of infile to get number of cities: `vertices`

If input is invalid then print error message and exit

Read the next `vertices` lines to get the city names

- Put each city in an array `cities`

- Remove the newline character at the end

Create graph, and make it undirected if specified

Read remaining lines of infile

- If line malformed, print error message and exit

- Otherwise, add each edge to graph

Create 2 paths: 1 for tracking current path and other for tracking shortest traveled path

Call `dfs`

```
void dfs( ... )
```

Description/Explanation

`dfs` will use recursion to find all hamiltonian paths. As it goes, it will compute and record path length, compute and record total number of recursive calls, select the shortest path, and write to file or stdout accordingly.

Pseudocode

```
1 procedure DFS(G,v):  
2   label v as visited  
3   for all edges from v to w in G.adjacentEdges(v) do  
4     if vertex w is not labeled as visited then  
5       recursively call DFS(G,w)  
6   label v as unvisited
```

Assistive code/abstract data types:

`graph.c`

Graph ADT

`path.c`

Path ADT

`stack.c`

Stack ADT