|  |
| --- |
| CIS 350 – Data Structures |
| Program 4 – Turn in 1 |
| Fall 2014 |

|  |
| --- |
| Daniel Frey  12-3-2014 |

Table of Contents

[1. Problem Summary 2](#_Toc405336759)

[2. Requirements Document 2](#_Toc405336760)

[2.1. Program Requirements 2](#_Toc405336761)

[2.2. Assumptions 2](#_Toc405336762)

[3. Decomposition Diagram 3](#_Toc405336763)

[4. Order 3](#_Toc405336764)

[5. Testing Strategy 3](#_Toc405336765)

[6. Test Plan – Version 1 4](#_Toc405336766)

[7. Initial Algorithm 4](#_Toc405336767)

[8. Test Plan – Version 2 5](#_Toc405336768)

# Problem Summary

Input from a file will contain multiple sets of data for graphs. Attempt to create a Hamiltonian cycle.

# Requirements Document

## Program Requirements

* For graph input
  + Take an input file of graph datasets
    - First line contains number of vertices followed by number of edges
    - Following lines contain edges
    - Vertices are integers numbered 1 to *n*
    - Edges are listed by vertices
    - Edges are numbered 1 to *m*
    - A space separates data in a line
    - Every input is a non-negative integer
  + Attempt to build a Hamiltonian cycle using backtracking
    - Start with vertex 1
* Output
  + Echo input
  + Echo vertices
  + Echo edges
  + Built graph as adjacency matrix/list
  + List sequence of actions
    - Add/remove vertex/edge
  + Print out cycle starting at 1 in lexicographic order

## Assumptions

* Graphs are simple
  + Contain no self-loops
  + Contain no parallel edges
* Data is valid

# Decomposition Diagram

# Order

* Graph
* Hamiltonian cycle
* Backtrack

# Testing Strategy

* Graph
* Hamiltonian cycle
* Backtracking

# Test Plan – Version 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Strategy Category | Test Number | Description | Input | Expected Result | Actual Result | Pass/Fail |
| File | 1.1 | File contains data |  |  |  |  |
| File | 1.2 | File doesn’t contain data |  |  |  |  |
| Graph | 2.1 | Read graph data from file |  |  |  |  |
| Graph | 2.2 | Read data set correctly |  |  |  |  |
| Graph | 2.2 | Edges added to graph |  |  |  |  |
| Hamiltonian cycle | 3.1 | Vertex added to cycle |  |  |  |  |
| Hamiltonian cycle | 3.2 | Edge added to cycle |  |  |  |  |
| Hamiltonian cycle | 3.3 | Graph with no edge has no cycle |  |  |  |  |
| Hamiltonian cycle | 3.4 | Disconnected graph has no cycle |  |  |  |  |
| Hamiltonian cycle | 3.5 | Graph has at least minimum number of vertices |  |  |  |  |
| Backtracking | 4.1 | Backtracking occurs |  |  |  |  |
| Backtracking | 4.2 | Backtrack to correct location |  |  |  |  |

# 

# Initial Algorithm

* Read file
  + While there is more to read from file
    - Read in number of vertices and edges
      * If vertices < 3, no Hamiltonian cycle possible
    - Build adjacency matrix graph of number of vertices
    - Read in edges for specified edge amount
    - Add edge to graph
* Hamiltonian Cycle
  + Start with vertex 1
  + Keep selecting the next unvisited minimum vertex
    - Backtrack if
      * A cycle is found before all vertices visited
      * A path is made including all vertices, but does not end where it started
  + For each new vertex picked
    - Compute the remaining unvisited vertices
  + End when all vertices are visited and a cycle beginning and ending with vertex 1 is made
* Backtracking
  + Use a structure like an array to store current path
  + Check that the vertex to be added is adjacent to the previous, and that it is not already included in the current path
  + If vertex does not lead to solution, remove it and try other paths

# Test Plan – Version 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Strategy Category | Test Number | Description | Input | Expected Result | Actual Result | Pass/ Fail |
| File | 1.1 | File contains data | File with data | Read and build graph |  |  |
| File | 1.2 | File doesn’t contain data | Empty file | Program stop |  |  |
| File | 1.3 | File exists | File name | Continue |  |  |
| File | 1.4 | File does not exist | File name | Ask for valid file |  |  |
| Graph | 2.1 | Read graph data from file | Graph file | Data read from file |  |  |
| Graph | 2.2 | Read data set correctly | Graph file | Read for set number of edges |  |  |
| Graph | 2.2 | Edges added to graph | Graph file | Edge added to matrix |  |  |
| Hamiltonian cycle | 3.1 | Path to next vertex via current edge | Graph | Vertex, edge added to path |  |  |
| Hamiltonian cycle | 3.2 | Graph with no edge has no cycle | Graph | No cycle, stop |  |  |
| Hamiltonian cycle | 3.3 | Disconnected graph has no cycle | Graph | No cycle, stop |  |  |
| Hamiltonian cycle | 3.4 | Graph has at least minimum number of vertices | Graph | If < fail, if >= try to find cycle |  |  |
| Backtracking | 4.1 | Backtracking occurs | Graph | Next vertex is already visited and does not form Hamiltonian cycle |  |  |
| Backtracking | 4.2 | Backtrack to correct location | Graph | Return to previous vertex, remove vertex and edge from path |  |  |