

# **Assignment 5**

## **Hamming Codes**

### **DESIGN DOCUMENT**

Zack Traczyk  
CSE13S - Spring 2021

Due: May 9<sup>th</sup> at 11:59 pm

## **1 Objective**

## **2 Given**

- Header files for stack, path, and graph
- Pseudocode for recursive search

## **3 Test Harness**

- h : Command line options
- v : Verbose printing; prints all Hamiltonian paths found as well as total number of recursive calls to dfs()
- i : infile; the input file containing the cities and edges of a graph (default should be stdin)
- o : outfile; the output file to print to (default is stdout)

### **3.1 Parse**

### **3.2 Execute**

## **4 Algorithm implementation**

```

increment calls counter;
if first call then
    | add vertex to path according to graph;
end
mark vertex as visited in graph;
edge number = adjacent_edges(graph, vertex to check, array to store in);
if path has hit every node then
    | push vertex to path;
    | if path is shortest then
    |     | if verbose argument set then
    |     |     | print path;
    |     | end
    |     | copy shortest path;
    | end
    | pop vertex from path;
else
    | for every adjacent vertex do
    |     | if edge not visited then
    |     |     | push adjacent vertex to path;
    |     |     | if if path length is not longer than shortest then
    |     |     |     | recursive call to DFS;
    |     |     | end
    |     |     | pop adjacent vertex from path;
    |     | end
    | end
end
add vertex to path according to graph;

```

**Algorithm 1:** DFS