

Assignment 5

Hamming Codes

DESIGN DOCUMENT

Zack Traczyk
CSE13S - Spring 2021

Due: May 9th at 11:59 pm

1 Objective

2 Given

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

3 Prelab Questions

a. Completed Lookup table

Table 1 on the following page shows which error vector values are associated with which bits/ error codes.

b. Decode the following codes. If it contains an error, show and explain how to correct it.

1110 0011 - $e = (1233) = (1011) = 1101 = 11 = \text{Bit 6}$ 1101 1000 - $e = (2121) = (0101) = 1010 = 10 = \text{HAM.ERR}$

4 Test Harness

- h : Command line options
- v : Verbose printing; prints all Hamiltonian paths found as well as total number of recursive calls to dfs()

Value	Bit
0	0
1	4
2	3
3	HAM_ERR
4	2
5	HAM_ERR
6	HAM_ERR
7	5
8	1
9	HAM_ERR
10	HAM_ERR
11	6
12	HAM_ERR
13	7
14	8
15	HAM_ERR

Table 1: Lookup Table

- `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`)

4.1 Parse

4.2 Execute

5 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