Fall 2003 **CMSC 311: Project 1** Kruskal and Scolnik

Due October 22, 11PM.

You are to write a C (or C++) program to determine the distance from START to FINISH in a maze. The maze is on an $m \times n$ grid (where $m, n \leq 1024$). Some grid points are walls, which are represented by a '1'. The other grid points are open space that can be moved onto, which are represented by a '0'. Distance is measured by moving one grid space north, south, east, or west. But you are not allowed to move into (or through) a wall. You can assume there is a path from START to FINISH.

INPUT:

- SIZE: Positive integers m, n.
- MAZE: An $m \times n$ table of bits, where a '1' represents wall and a '0' represents open space. Each row will start on a new line. The bits will be stored in base 10 unsigned numbers. The first number in a row will represent the first 32 bits of the row, the second number will represent the second 32 bits, etc. You can assume $32|n$.
- START: An $m \times n$ table of bits, where a '1' represents the start location and all other bits are '0'. It it represented the same way as MAZE.
- FINISH: An $m \times n$ table of bits, where a '1' represents the finish location and all other bits are '0'. It it represented the same way as MAZE.

(MAZE, START, and FINISH should each use $mn/32$ words of memory.)

ALGORITHM:

```
1. {Produce the complement of MAZE; this is the OPEN space that can be moved onto.}
   OPEN ← not MAZE.
2. NOW ← START;    COUNTER ← 0;
3. {Check if arrived at FINISH.}
   If NOW & FINISH ≠ 0 then print COUNTER and exit;
4. Increment COUNTER;
5. {Find new locations you can reach: For every '1' bit in NOW copy it north, south,
   east, and west into NEW_NOW. Do this using bit operations on words. Be careful
   about crossing word boundaries east and west.}
6. {Make sure you do not walk into a wall.}
   NOW ← NEW_NOW & OPEN;
7. Go to step 3;
```

Sample input files will be provided in the posting account directory ss311001/Projects/P1.