Algorithms
Graph Modeling Project (125 pts)

A detailed description of the maze problem including a figure is provided separately[1]. You may want to look at that first to get a sense of what this project involves! Your assignment is to (1) model the maze problem as a graph (2) use an appropriate graph algorithm that we've encountered in class to solve the problem (3) code the algorithm and (4) provide the output on an example input. **Friendly reminder: This is an individual assignment. The departmental collaboration policy (the document you signed at the start of the semester) will be enforced.**

1 Deliverables

The deliverable will take the form of a report (a printed copy). If we are not confident your approach works, we may follow up and ask for a demo. Your report must include the following four items.

1.1 Problem Modeling[50 pts]

- [10] Explain how you modeled the problem as a graph (typically, a few sentences)

- [15] Draw enough of the resulting graph to convince us that you have modeled the graph correctly (you don't have to draw the whole graph).

- [10] Identify the graph algorithm needed to solve the problem (one sentence).

- [15] Convince us that this algorithm will actually solve the problem. This means that it will find a path if one exists (typically, a few sentences, depending on your approach).

1.2 Code Submission [30 pts]

Please include your code in the report. 10 points will be allocated for quality and readability of the code. The code should read in the provided input file and print the output.

1.3 Results [45 pts]

Display the output of your program on the input maze provided below. Points will be awarded for finding a correct path and printing it in the required format.

1.4 Extra Credit [10 pts]

You will receive a bonus of upto 10 points if you can show that you have done something above and beyond what was asked. Possible examples might be (1) using an algorithmic library (e.g., Boost in C++) to implement the graph functionality (2) augmenting your algorithm with a visualization (3) an experimental comparison between algorithms etc. *Please explicitly include a section in your report titled "Extra Credit." Don't just assume the TA will notice your enhancements.*

---

[1]The problem was taken from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People," by Robert Abbott, Bob Adams, Inc. Publishers, 1990.

2   Input

The first line contains an integer n representing the number of bugs (not including the two spiders). For the example provided, n is 91. Each bug has an integer ID between 1 and n. IB Spider is assigned integer ID 0 and Lady Spider is assigned ID $n + 1$ (92 in our example). Each subsequent line describes a "web link": it includes the IDs of the two bugs and the direction of the link using cardinal directions (E = east, NW = northwest, etc). For example, the next input line indicates that the bug at ID 0 (IB Spider) has an east-going link to the bug at ID 1.

```
91
0 1 E
1 2 E
2 3 E
3 4 E
4 5 E
5 6 E
6 7 E
7 8 E
8 9 E
9 10 E
10 11 E
1 12 S
12 19 S
3 14 SW
14 13 SW
13 19 SW
3 15 S
15 20 S
10 16 S
16 17 S
17 27 S
11 18 S
18 28 S
20 21 E
21 22 E
22 23 E
23 24 E
24 25 E
25 26 E
26 27 E
27 28 E
19 29 S
29 33 S
29 34 SE
20 30 S
30 35 S
27 31 S
```

```
31 42 S
31 41 SW
28 32 S
32 43 S
28 31 SW
33 34 E
34 35 E
35 36 E
36 37 E
37 38 E
38 39 E
39 40 E
40 41 E
41 42 E
42 43 E
33 44 S
44 50 S
35 45 S
45 51 S
34 45 SE
45 52 SE
41 46 SW
46 47 SW
47 54 SW
42 48 S
48 57 S
43 49 S
49 58 S
49 57 SW
52 53 E
53 54 E
54 55 E
55 56 E
56 57 E
50 59 S
59 63 S
51 60 S
60 64 S
52 61 SE
61 65 SE
57 62 SW
62 66 SW
57 68 S
58 69 S
66 67 E
67 68 E
68 69 E
63 74 S
```

```
64 76 S
63 70 SE
70 76 SE
65 71 SE
66 71 SW
71 79 SW
71 80 SE
68 72 S
72 81 S
69 73 S
74 75 E
75 76 E
76 77 E
77 78 E
78 79 E
79 80 E
74 82 S
76 84 S
76 85 SE
79 86 SW
80 89 SE
81 90 S
73 91 S
82 83 E
83 84 E
84 85 E
85 86 E
86 87 E
87 88 E
88 89 E
89 90 E
90 91 E
84 92 S
```

3   Output Format

The output should consist of a path from ID 0 to ID $n + 1$, The path should be described as a sequence of IDs. For example, one possible path may start off as follows

```
0 3 19 44 63 85 ...
```