Algorithms
Graph Modeling Project (125 pts)

This problem is based on a maze problem (from "MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People," by Robert Abbott, Bob Adams, Inc. Publishers, 1990). The description of the problem including a figure is provided separately. You're welcome to try figuring out the solution to the puzzle on your own, but that won't get you any points! Your assignment is to model the maze as a graph and to solve the problem using an appropriate graph algorithm that we've encountered in class. Next, write a program to solve the problem that uses one of the algorithms we covered in Chapter 5 of the textbook. You are encouraged (but not required) to use a library implementation of the graph algorithm.

Important: Note that this is an individual assignment. The departmental collaboration policy (the document you signed at the beginning of the semester) will be strictly enforced.

DELIVERABLES

1. **Problem Modeling[50 pts]**:
    - [10] Explain how you modeled the problem as a graph.
    - [15] Draw enough of the resulting graph to convince us that you have modeled the graph correctly.
    - [10] Identify the graph algorithm needed to solve the problem.
    - [15] Argue that this algorithm will actually solve the problem.
2. **Code [30 pts]**: Your code must read the input from a file: the first line of this file contains an integer representing the number of villages followed by a second integer that contains the number of transit lines (edges). For the example provided, there are 36 villages labeled $A, B, C, \ldots, X, Y, Z, a, b, \ldots, i, j$ and 70 transit lines. Each subsequent line describes a transit line: it includes the two villages joined by the line, the color of the line (R, G, B) and the *type* of the transit line (H, C, T, B) for Horse, Cable car, Trolley, and Bus, respectively. For example, the next input line indicates that the first transit line joins villages A and B, using the Red company, and that this is a Cable Car. The entire input file is included at the end of this document.
3. **Results [45 pts]**: Display the output of your program on the maze provided. The output should consist of a path from Startsburg (Village A) to Endenville (Village j). The path should be described as a sequence of village ids. For example, one possible path may start off as follows

    ```
    A B F K P Q R
    ```
4. **BONUS [10 pts]**: You will receive a bonus of 10 points if your implementation uses an appropriate graph function call from an algorithmic library (e.g., LEDA or Boost). LEDA is available on the alamode machines (in the Linux Lab in BB). Information about LEDA (including some sample code) is available on blackboard.

```
36 70
A B R C
B E B C
B C B T
C D G T
D J R T
B F R T
C G R C
D G B B
D I G B
E F G H
F G G H
G H R C
H I R H
I J B H
E O R C
E K G B
F K R H
G K B B
G L G T
H L R B
I M G T
I N G H
J N R B
L M B B
M N B H
K O B T
K P R C
L Q G T
L R R H
M S R T
N T R B
P Q G C
Q R B C
R S B H
S T B B
O U R B
O V B H
P V G H
Q W R T
R X R T
S X G B
S d R H
S Y G C
T Y G T
U V G B
V W R C
W X G C
```

```
U Z B B
V Z R C
V a G B
V b B T
W b R T
X c R H
Y e G T
Z a R T
a b R C
b c G T
c d G B
d e G H
Z f B H
a f G H
b g R T
b h B B
c h R C
d i R H
e i R B
f g B C
g h G C
h i R B
i j B B
```