

Will B
Mr. Yeh
AT Computer Science
12/10/2015

Documentation for Rankings Project

Overview:

This project is a system that computes rankings from a field of competition containing head-to-head matchups. The field I used for my project was the 2015 Southeastern Conference (SEC). This program takes in data from head-to-head matchups (from real results found online) in the form of a formatted text document. Those matchups are used in the creation of team nodes, which contain data about each team, including its wins, losses, scores, and more. The teams are then ranked using the topological sort algorithm if possible. If not possible (in the event of tie-breaking/chains), the teams are ranked according to a “point sort” algorithm. This “point sort” algorithm traverses through the graph of each team’s losses, assigning points to a team every time its node is visited.

File Input:

The user can implement any data set needed, provided that it contains matchups containing two teams with scores. There is also space for extra information, such as date and time, in the input but the program doesn’t use this information in any way. To enter data, follow the following steps.

1. Open a simple text file (must end in “.txt”)
2. Input your matches as follows:

Match Number , Date , Time , Day of Week , Winning Team , Winning Team Score , Blank , Losing Team , Losing Team Score

As mentioned above, all you need are the Winning Team and Losing Team, and their scores. If the user doesn’t want to enter some of the optional data they can use this example:

Blank , Blank , Blank , Blank , Winning Team , Winning Team Score , Blank , Losing Team , Losing Team Score

3. Go to line 1 of the program and change the "FILE_TO_READ" string to the name of your file.

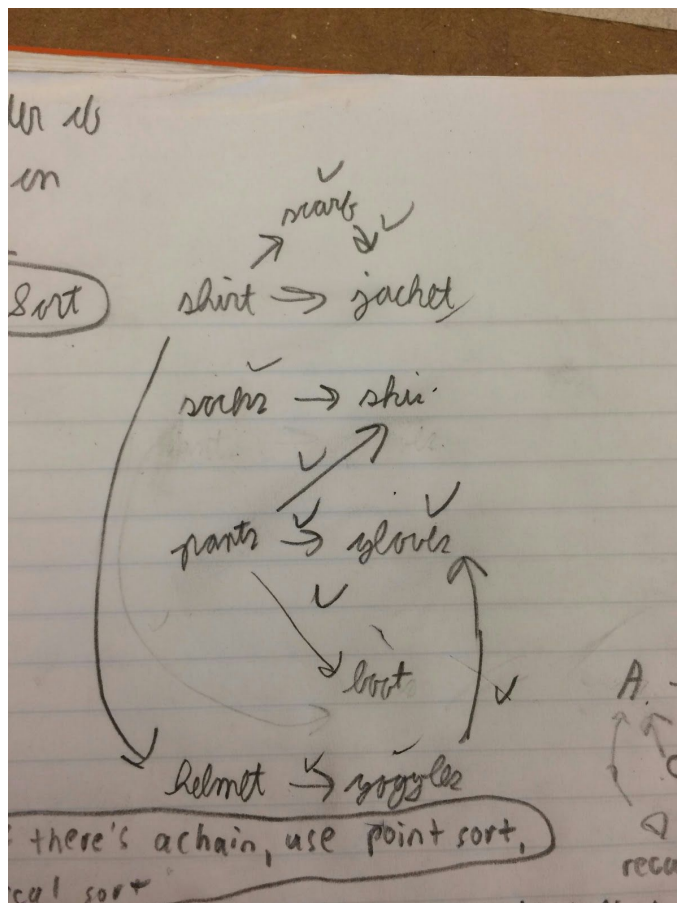
```
FILE_TO_READ = "Pac12.txt"
INFO_LINE_INDICATOR = "Rk"
MATCH_NUM = 0
WINNER = 5
WINNER_POINTS = 6
LOSER = 8
LOSER_POINTS = 9

DROP_OUT_MODIFIER = 1
POINT_SORT_TEST_AMOUNT = 10000000
```

Test Cases:

Provided as examples are three test cases.

1. "Ski Test 1.txt" This test case is an example of a chain of $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$. The program recognizes the chain during topological sort and moves to point sort.
2. "Ski Test 2.txt" This test case is the one discussed in class. It contains no chains and thus it uses the topological sort algorithm.



3. "2015 Southeastern Conference.txt" This test case is the normal case. It contains chains and thus uses the "point sort" algorithm.

To be Noted:

1. If there is a tie-game in one of the matches, the match will not be counted as a win or loss for either team.

2. To change the amount of trials used by "point sort", go to line 10 of the program and change the POINT_SORT_TEST_AMOUNT integer.

```
FILE_TO_READ = "Pac12.txt"  
INFO_LINE_INDICATOR = "Rk"  
MATCH_NUM = 0  
WINNER = 5  
WINNER_POINTS = 6  
LOSER = 8  
LOSER_POINTS = 9
```

```
DROP_OUT_MODIFIER = 1  
POINT_SORT_TEST_AMOUNT = 10000000
```

3. The text files may need to be on the desktop for them to be recognizable to the program.