CSE 231, Spring 2008
Programming Project 6

Assignment Overview

This assignment is worth 40 points (4.0% of the course grade) and must be completed and turned in before 11:59pm on Monday, Feb 25th, 2008. The purpose of this project is to work more with lists, looping, and file I/O. For this assignment, you will work with real data from the web to find the best players in NBA history.

Background

The National Basketball Association (NBA) is the USA's premier professional men's basketball league. It has 30 teams; 29 in the United States and one in Canada. It is an active member of USA Basketball (USAB), which is recognized by the International Basketball Federation as the National Governing Body (NGB) for basketball in the United States. The NBA is one of the four major North American professional sports leagues which are NBA, NHL, NFL and MLB.
Now you are going to write a program to find the best players in NBA history.

Program Specifications

You are to get the raw NBA stats from the Web, compute the efficiency of 3924 players, and output the 50 most efficient players, as well as some other interesting statistics.

Your program is to take no input (except for reading the file) and produce a file that contains all the results.

NBA Data Collection and Manipulation

1. Goto http://www.databasebasketball.com/stats_download.htm
2. Download databaseBasketball2.1.zip
3. Unpack the zip file (most machines will do this when you double-click the file)
4. We will only do statistics based on "player_regular_season_career.txt". So copy this file into the same directory where you write your python program.

The format of this file is easy to understand. The first line tells you the names of all columns. From the second line, each line's data corresponds to one player's career regular season's statistics. Each field is separated by "|"

To understand the meanings of each of the abbreviations, look at the page:
http://www.databasebasketball.com/about/aboutstats.htm

Efficiency
How do many NBA coaches quickly evaluate a player's game performance? They check his efficiency.

NBA.com evaluates all players based on the efficiency formula indicated below (and shown on the aboutstats.htm page)

In this project, we will follow this efficiency formula. Since we are not evaluating a player based on one game, we need to divide the total efficiency by the number of games the player played.

So the formula is:

$$Efficiency = \frac{(pts + reb + asts + stl + blk) - ((fga - fgm) + (fta - ftm) + turnover)}{gp}$$

The technical words on the right side correspond to the fields in the statistics file.

Other Stats

Besides efficiency, also collect:

1. The player who played the most minutes
2. The player who played the most games
3. The player who scored the most points
4. The player who got the most rebounds
5. The player who got the most penalties
6. The player who made the most freethrows

Your high level algorithm will be:

1. Read raw data line by line from “player_regular_season_career.txt”
    a. For each player, calculate his efficiency.
    b. Put player’s efficiency, first name, and last name, into a list.
    c. collect the ‘top 6’ for the values listed in ‘Other Stats’
2. Print the Other Stats list (last name, first name, value) for each of the categories listed above. Provide a description, something like:
    a. Most points scored: Magic Johnson, 12345
3. Sort the list according to players efficiency
4. Output the top 50 players’ first name, last name along with efficiency into a file “top50.txt”. Each line corresponds to one player.

Deliverables

You must use handin to turn in two files: top50.txt, and proj06.py – this is your source code solution; be sure to include your section, the date, the project number and comments describing your code. Please be sure to use the specified file names, and save a copy of your proj06.py file to your H drive as a backup.

Assignment Notes

1. When read the input file, you should be careful about the first line which does not contain any data.
2. Don't forget to convert the string to number.

   Sample Usage:

   ```
   S = "123"

   I = int(S)

   S = "12.3"

   F = float(S)
   ```

3. Since there are so many fields, do some testing (E.g. output some parsed data) to make sure that you get the correct data.
4. List's sort function and reverse function should be useful.

   ```
   Li = [ [3,2], [1,2], [2,5]]

   Li.sort()    # Li will be   [ [1,2], [2,5], [3,2]]

   Li.reverse()   # Li will be [ [3,2], [2,5], [1,2]]
   ```

5. To open a file for output, remember:

   ```
   fileDescriptor = open('fName.txt','w')

   fileDescriptor.write('stuffToWrite')

   fileDescriptor.close()
   ```