

Computing Odds for Sports competitions

Disclaimer: The algorithm described below is only a mathematical abstraction, and cannot be expected to work (and should not be used) in any real-life scenarios.

1 Project description

In a variety of sports (e.g. the final phase of the Champions League in soccer) teams are scheduled in a tree.

- The play takes place in rounds. In each round surviving teams are matched in pairs according to the tree structure.
- Two matched teams play each other, the winning advances to the next round.
- The initial number of teams is a power of two (e.g. 32). The number of surviving teams gets halved in each round (e.g. 16 then 8 then 4 then 2), until the final, where the team that wins that game wins the whole competition.

Various sports use a point-based system to evaluate the “strength” of a team. Stronger teams have a higher number of points (in European soccer this is given by the UEFA Rankings).

For this project, you will implement a program that will evaluate the winning odds in a tree-based tournament, based on team strengths.

2 What the program should do

The program should read a set of teams and team strengths from a file, and should compute and output the odds of various teams winning a tree-based tournament.

The odds of a given team winning the competition is the number of team wins in 100 random simulations of the tournament, divided by 100.

We assume that, whenever two teams A and B of strengths a and b meet, the first team wins with probability $\frac{a}{a+b}$. You can implement this choice by choosing a random number in the range $(0, a + b)$. If the random number is larger than a then the second team wins, otherwise the first team wins.

To compute the advancing teams you should use one *non-recursive* version of a tree-traversal methods we learned in class (you have to decide which one is appropriate).

The initial allocation of teams to the leaves of the tree should be random. **Please note that the program should work with an arbitrary number of teams (that is a power of two)**, not only 16, as in the example below.

3 What to turn in

You will turn in a report containing:

- A description of the significant choices/issues in the design of your code and experiments.
- The results of your experiments, followed by a few lines describing your conclusions.
- The listing (source code) of your programs.

You may turn in the document in class, or via email (gabriel.istrate@gmail.com).

4 Coding standards

A percentage of your grade will be based on the quality of your code, so pay attention to it. Discuss changes (if any) you made to programs presented in class. Take extra care in documenting the code you are implementing on your own. Properly modularize the code (for instance implement separate functions for significant parts of the program).

5 Deadline

Two weeks from now (May 5, 2008, 4PM local time). This is a strict deadline. No credit will be given for homework turned in late.

6 Test case

You can use a file with the following contents as a testcase for your program (to help reading from the file, I have represented team names using a single string). For your convenience, the test file is available from the course webpage as well.

Shalke04 60.640
FCPorto 89.107
ASRoma 78.808
RealMadrid 104.374

Olympiakos 42.415
Chelsea 99.618
Liverpool 112.618
Internazionale 107.808
Celtic 62.064
FCBarcelona 119.374
Lyon 95.706
ManchesterUnited 99.618
Fenerbahce 36.791
Sevilla 87.374
Arsenal 104.618
Milan 133.808