

# HW2 FaceMash and MapReduce

## Goals

In this assignment, you will use the python programming to implement the Elo rating algorithm which was used in FaceMash and use map-reduce programming to implement matrix multiplication.

## Elo Rating

The Elo rating system is a method for calculating the relative skill levels of players in competitor-versus-competitor games such as chess. The Elo system was originally invented as an improved chess rating system, but is also used as a rating system for multiplayer competition in a number of video games,[1] association football, American football, basketball,[2] Major League Baseball, Scrabble, board games such as Diplomacy and other games.

The difference in the ratings between two players serves as a predictor of the outcome of a match. Two players with equal ratings who play against each other are expected to score an equal number of wins. A player whose rating is 100 points greater than their opponent's is expected to score 64%; if the difference is 200 points, then the expected score for the stronger player is 76%.

A player's Elo rating is represented by a number which increases or decreases depending on the outcome of games between rated players. After every game, the winning player takes points from the losing one. The difference between the ratings of the winner and loser determines the total number of points gained or lost after a game. In a series of games between a high-rated player and a low-rated player, the high-rated player is expected to score more wins. If the high-rated player wins, then only a few rating points will be taken from the low-rated player. However, if the lower rated player scores an upset win, many rating points will be transferred. The lower rated player will also gain a few points from the higher rated player in the event of a draw. This means that this rating system is self-correcting. A player whose rating is too low should, in the long run, do better than the rating system predicts, and thus gain rating points until the rating reflects their true playing strength.



## Problems

In this assignment, you will need to solve 4 problems. Please follow the instructions in the problem files.

- Problem 1: 30 points (10+10+5+5 points)
- Problem 2: 20 points (5+5+10 points)
- Problem 3: 20 points (10 points for each subquestion)
- Problem 4: 30 points (10 points for each subquestion)

## Debugging your program using python unit tests (nosetests)

You can use the provided unit tests to debug your code. For example, to debug the code of your solution to problem 3, type:

```
nosetests test3.py
```

If the code is correct, you will pass the unit tests.

## How to submit?

Please submit problem[1-6].py (six python files) in Canvas system under “HW2 assignment”.