Project Name: The Basketball elimination problem

By: Alon Lapid, **Github account :** https://github.com/Pomona-College-CS181-SP2020/NBAelimination

Project description:

The basketball elimination problem is defined as the ability to determine, in the middle of the basketball season, which teams could **not** finish in first place in their conference (east, or west), and eliminate these teams from the list of potentially winning teams. This determination will be based on the current standing (at a certain point in time) of the teams' game results, and the remaining games that each team is scheduled to play during that season. For each team, all possible scenarios of game outcomes will be evaluated to determine if the team is eliminated or not. I used the historical NBA data from https://fixturedownload.com/results/nba-2018 and tested at multiple time points during the season to evaluate which teams are eliminated. The challenge is to make the elimination determination run in polynomial time and therefore a simple "brute force approach" will not work as it runs in exponential time. The project utilizes a methodology from graph theory known as the max-flow problem to solve the elimination problem efficiently.
The final presentation provides an in-depth explanation of how the algorithm works.

Project design:

The project is divided into 3 main libraries:

Maxflow library – a Library the solves the max flow problem using the Edmund-Karp algorithm given the network flow graph.

Games library – a Library that provides useful utilities over a list of games, for example computing the current standings, getting the maximum possible points for each team and so on.

Elimination library – a Library that converts the elimination problem into a maxflow problem, solves the maxflow problem and determines for each team whether they are eliminated. It utilizes the previous two libraries.

Input and output:

The input for the project is a games file that includes the result of the already played games with their results + the games to be played and a file that lists the teams and the conference they belong to. The output of the project is a list of all eliminated teams.

Usage example:

```
downloadGamesfromWeb  "https://fixturedownload.com/download/nba-2019-EasternStandardTime.csv" "test/nba-2019-EasternStandardTime.csv"

elimination_nba_2019 <- eliminationMaxFlowFromFile  "test/teamsnba.csv"  "test/nba-2019-EasternStandardTime.csv"
```

Testing:

One of the challenges in this project is to be able to test and validate results. Therefore, I picked small samples that can be solved using brute force method in order to compare these results to the results of my algorithm. I also picked well known examples from the internet and showed that my algorithm yields identical result. All tests are in test\Spec.hs

Project results against goals:

The project met its designed goal. We tested it on the 2019-2020 NBA suspended season and it appears to work well.

Possible project extension:

The main challenge in this project is testing on large-scale examples where the brute force approach for testing is impractical. One promising option, that I did not have time to implement, is to try to solve the maxflow problem as a set of linear constraints. This can be solved efficiently by the simplex method and so it could provide another way to sanity check the results.