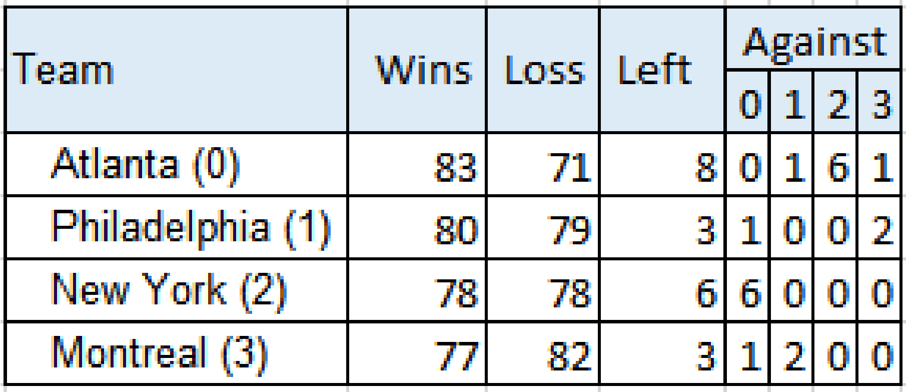
EECS 278: Advanced Algorithms Project Report
Team members : Rahul Hoskeri, Asoke Datta

Baseball Elimination Given the standings in a sports league at some point during the season, determine which teams have been mathematically eliminated from winning their division.

In the baseball elimination problem, there is a league consisting of N teams. At some point during the season, team i has w_i wins and r_{ij} games left to play against team j. A team is eliminated if it cannot possibly finish the season in first place or tied for first place. The goal is to determine exactly which teams are eliminated.



Team	Wins	Loss	Left	Against			
				0	1	2	3
Atlanta (0)	83	71	8	0	1	6	1
Philadelphia (1)	80	79	3	1	0	0	2
New York (2)	78	78	6	6	0	0	0
Montreal (3)	77	82	3	1	2	0	0

Figure 1: Sample Point table

Montreal is mathematically eliminated since it can finish with at most 80 wins and Atlanta already has 83 wins. **Philadelphia** is also mathematically eliminated. It can finish the season with as many as 83 wins, which appears to be enough to tie **Atlanta**. But this would require **Atlanta** to lose all of its remaining games, including the 6 against **New York**, in which case **New York** would finish with 84 wins. We note that **New York** is not yet mathematically eliminated despite the fact that it has fewer wins than **Philadelphia**.

Input:

First line of input consists of N, total number of teams.
The next N lines consists of team information ”team id”,”wins”,”loss”,”games to be played” and ”with whom”.
The Team id are numbered 0,1,2,...N-1. Figure 1 describes a sample input for required program.

Output:

Output will be team names {0,1...N-1}, separated by space. e.g. 0 2 if 0 and 2 are the teams eliminated.
Output is ”-1” if no team is eliminated.
Output for Figure 1 :
1 2

Assumptions. Assume that no games end in a tie,no rainouts, Ignore wildcard possibilities,assume that there are no whitespace characters in the name of a team.

Project Flow Diagram:

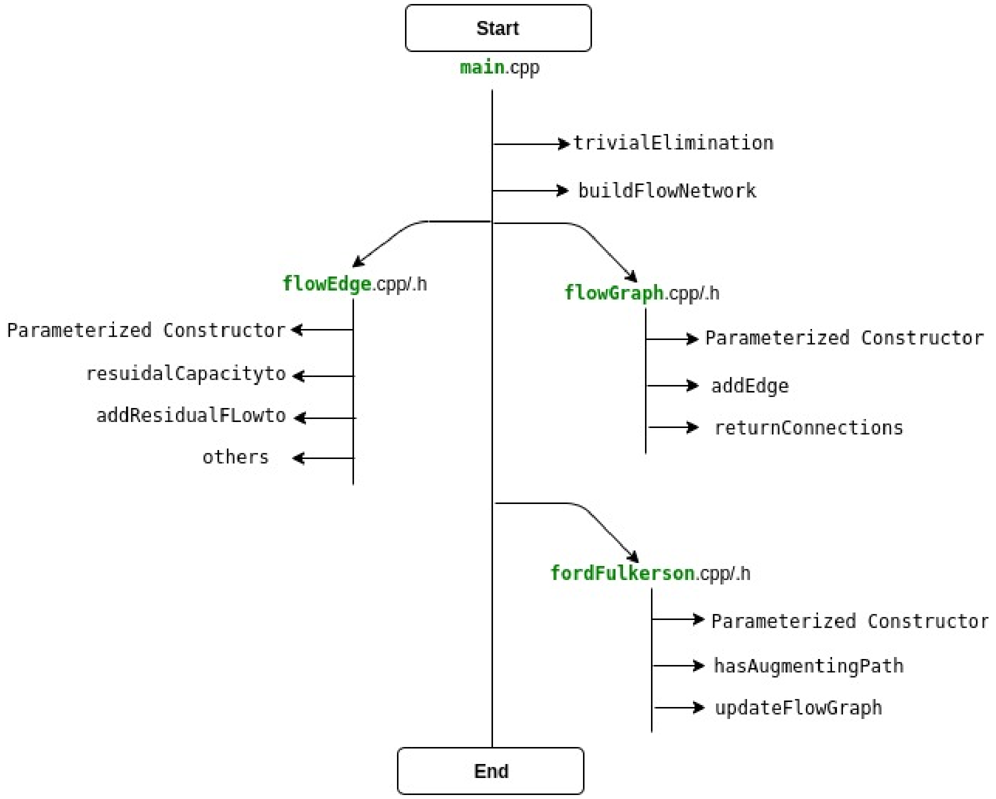


Figure 2: **Project Flow Diagram**

Project Details:

1. **main.cpp**
   - Load the input file.
   - Check for trivial elimination.
   - Build flowgraph for teams which are not trivially eliminated.

   (a) Create a source(Number of teams +1) and sink(Number of teams +1)

   (b) Create gameNodes(Which team is playing with whom) and connect with source. Each connection is represented as a **flowedge.**

   (c) connect gameNode with all the teams except one(For which we are building flowGraph).

   (d) Construct flowGraph(Its a data structure which represent the whole graph) using the edges from last step.

   (e) Call Fordfulkerson algorithm.

(f) In updated flowGraph check if all the **edges from source to gamenode are saturated or not. If all edges are saturated then the team is not eliminated otherwise eliminated**.

2. **Fordfulkerson.cpp**

   (a) Check for augmenting path. If any, update the flowGraph with bottleneck value.

   (b) Repeat the process until there is a augmenting path.

3. **flowGraph.cpp**

   (a) Store the full network as a adjacency list(Which is 2D vector in our implementation).

   (b) Supports adding new edge and returning all edges for any vertex.

4. **flowEdge.cpp**

   (a) Store edge information.

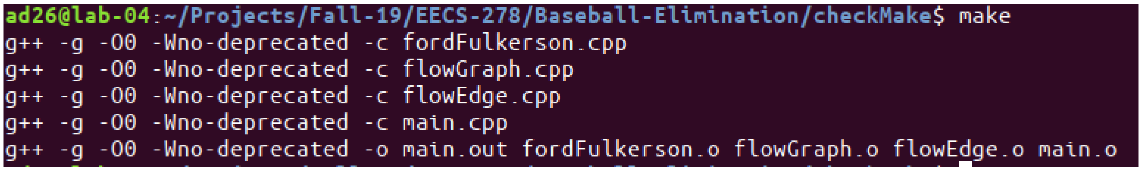
   (b) Supports adding residual capacity,flow.

Evalution :

We have tested our program in different input cases including edge cases also. Our program returns -1 if there is no team eliminated. Source code and evaluated input is added with the report.

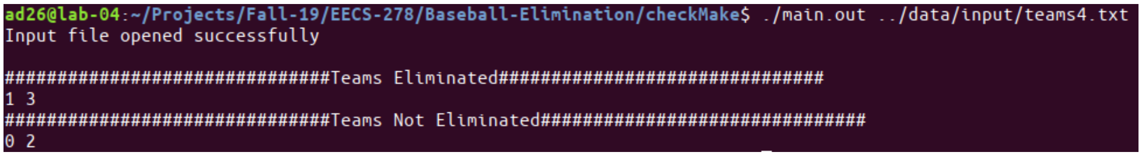
How to run the program :

1. run **Make**. It will create a executable **main.out** file.



```
ad26@lab-04:~/Projects/Fall-19/EECS-278/Baseball-Elimination/checkMake$ make
g++ -g -O0 -Wno-deprecated -c fordFulkerson.cpp
g++ -g -O0 -Wno-deprecated -c flowGraph.cpp
g++ -g -O0 -Wno-deprecated -c flowEdge.cpp
g++ -g -O0 -Wno-deprecated -c main.cpp
g++ -g -O0 -Wno-deprecated -o main.out fordFulkerson.o flowGraph.o flowEdge.o main.o
```

2. Run the executable with input file.



```
ad26@lab-04:~/Projects/Fall-19/EECS-278/Baseball-Elimination/checkMake$ ./main.out ../data/input/teams4.txt
Input file opened successfully

###############################Teams Eliminated###############################
1 3
###############################Teams Not Eliminated###############################
0 2
```

Reference :

1. https://www.cs.princeton.edu/courses/archive/spr03/cs226/assignments/baseball.html

2. https://github.com/nastra/AlgorithmsPartII-Princeton/tree/master/baseball