

Programming Assignment 3: voting (due March 4 before 5pm)

Please read the rules for assignments on the course web page (<https://courses.cs.duke.edu/spring22/compsci323d/>). Use Ed Discussion for questions and Gradescope to turn this in. Please use clear variable names and write comments in your code where appropriate (you can put comments between `/*` and `*/`, or start a line with `#`).

Please see Homework 1 for details about getting set up with GLPK, making a directory for this homework, etc.

Your submission should be a PDF describing your solutions. It should also contain the code and output of your code for the last problem, on the example given there. Please make sure that the code can be easily copied from the PDF. In particular, please do not just insert a screenshot of your code.

For the purpose of this assignment, we made up a new voting rule that we will call “ModifiedBorda.” An alternative j ’s score under ModifiedBorda is: the smallest number of votes that need to be removed so that j becomes the Borda winner (so lower ModifiedBorda scores are better). Actually, being tied for the Borda win is enough; so really, the goal is to compute the smallest number of votes that need to be removed so that there is no longer any other alternative j' that has a strictly higher score than j . Hence, an alternative’s ModifiedBorda score is never larger than the total number of votes, because if we remove all votes, then all alternatives are tied for the win.

Consider the following example with the following four votes:

1. $a \succ c \succ b$
2. $a \succ c \succ b$
3. $b \succ a \succ c$
4. $b \succ c \succ a$

The regular Borda scores of the alternatives are 5 for a , 4 for b , and 3 for c . The ModifiedBorda score of a is 0, because a is one of the Borda winners without removing any votes. The ModifiedBorda score of b is 1, because by removing (e.g.) the first vote, the Borda scores become 4 for b , 3 for a , and 2 for c . The

ModifiedBorda score of c is 2, because no single vote can be removed to make c have at least as many Borda points as a (no vote gives a 2 more Borda points than c), and removing (e.g.) the second and third votes will result in Borda scores of 2 for all alternatives. In this particular example, it so happens that the ModifiedBorda ranking of the alternatives is the same as the original Borda ranking: a wins, b is second, and c is third.

1. (10 points.) Give a quick argument why the winning alternative(s) will always (in any example) be the same under Borda and ModifiedBorda.

2. (20 points.) Give an example, i.e., a set of votes, in which the Borda ranking and the ModifiedBorda ranking of the alternatives are *not* the same, even though the winners (top-ranked alternatives) are. (If one of the rules produces a tie in the ranking and the other does not, we will count that as not the same, so that's enough. Try to make your example as small as possible—it can be very small!)

3. (70 points.) Create an integer program for calculating the ModifiedBorda score of a single candidate as a .mod file. To make things easy: We will think of the alternative for which we want to calculate the score as special, so we will distinguish between that alternative and the “other” alternatives. We will also assume that we are explicitly given as input the Borda score that each alternative receives from each vote. For vote i and the special alternative, we will refer to the Borda score that the special alternative gets from i as

`special_score[i]`

For vote i and some other alternative j , we will refer to the Borda score that j gets from i as

`other_score[i,j]`

For example, for the votes above and treating c as the special alternative (i.e., the one for which we would like to calculate the score), the data part of the file should look as follows:

```
data;
set VOTES := v1 v2 v3 v4;
set OTHER_ALTERNATIVES := a b;
param special_score := v1 1 v2 1 v3 0 v4 1;
param other_score: a b :=
    v1 2 0
    v2 2 0
    v3 1 2
    v4 0 2;
end;
```

You should also test your code on your own example from part **2**. (But turn in the output resulting from the above example `data;` part.)