CS 5110/6110 Program 3: Social Choice (40 points)

The Problem

Arrow's impossibility theorem says that there is no social choice mechanism that takes individual preference patterns and generates a fair societal preference pattern. Arrow defined fairness according to axioms, and showed that all the axioms could not be simultaneously satisfied. Voting methods are attempts to take individual preference patterns and create a "fair" societal preference pattern. We should be able to identify situations where the voting mechanism breaks down. Since there is no way for voting to be fair, the task of the designer is to minimize the unfairness.

In addition, the well-known Gibbard-Satterthwaite theorem tells us that no voting mechanism except dictatorship of one individual is strategy-proof on an unrestricted preference domain provided that there are at least three alternatives. An outcome manipulation (strategy) aims to bring about an outcome with a higher payoff than the outcome resulting from sincere behavior.

Consider a domain where voting may be appropriate. You can adapt the form of the input to reflect that domain. Consider a society of V voters who are voting on C candidates. Each agent ranks the candidates as 1 (meaning the best) and |C| (meaning the worst). In case two or more alternatives tie for the highest score, the winner is determined by a fixed tie-breaking rule, say in alphabetical order.

To make displaying easier, if multiple voters agree on the same ordering, we show it as an “occurs” count.

Data Set 1:

Choice/Voters	**A (occurs 5)**	**B (occurs 4)**	**C (occurs 3)**	**D (occurs 6)**
Alex	3	1	2	4
Bart	1	2	5	7
Cindy	2	3	7	3
David	4	7	6	6
Erik	6	5	3	1
Frank	5	6	1	2
Greg	7	4	4	5

Data Set 2.

Choice/Voters	**A (occurs 5)**	**B (occurs 4)**	**C (occurs 3)**	**D (occurs 6)**
Alex	2	1	2	4
Bart	3	3	3	1
Cindy	1	4	4	2
David	4	2	1	3

Allow the matrix to re-populated with values in each of the following ways:

- The specific data values shown in Data Set 1 shown above.
- The specific data values shown in Data Set 2 shown above.
- User may enter own occurrences and preferences
- Random preferences and occurrences can be used.
- All voters have the same preference between two specific choices (let user specify a relationship which all voters adhere to, like Bart > David). Did your voting mechanisms preserve this order in the final ranking? (Pareto Optimal)
- Another option you consider interesting.

Your job is to help these agents (as a group) choose one of the alternatives from the set of candidates. If your method gives a complete ordering, show it.

The Experiment

(a) Display the majority graph representing the votes for the candidates. (This can be in table format if you prefer.)

(b) Determine what choice is made when various social choice functions are applied. You do not need to do all of these. Pick three you consider interesting.

1. Slater Ranking: find an ordering (no cycles) which has the fewest disagreements with the majority graph. In output, label choices in Slater Ranking as 1, 2, …7.
2. Kemeny ranking; create an overall ranking of the candidates that has as few *disagreements* as possible (where a disagreement is with a ranking on a pair of candidates). In output, label choices in Kemeny Ranking as 1, 2, …7.
3. Bucklin ranking: start with k=1 and increase k gradually until some candidate is among the top k candidates in more than half the votes; that candidate wins. Identify the winner and the k required.
4. Identify the winner using the Second order Copeland technique.
5. Single Transferable Vote (STV, aka. Instant Runoff): candidate with lowest plurality score drops out; if you voted for that candidate (as your first choice), your vote transfers to the next (live) candidate on your list; repeat until one candidate remains. In the output, the one listed as choice 7 is the first to drop out.

(c) Once the results are in, can any voter (or collection of voters with the same ranking) vote strategically to get an advantage?

The Output:

Allow the user to see the results for individual voting methods and easily compare the results. I suggest you arrange the results something like the following

Ranking of choice (1 is best)	Slater	Kemeny	Bucklin K=2
1	Greg	Alex	Alex
2	Bart	Bart	XX
3	Cindy	Cindy	XX
4	David	David	XX
5	Erik	Greg	XX
6	Frank	Frank	XX
7	Alex	Erik	XX

Turn in a video clip of your output in which you demonstrate interesting results of the methods you compared.

Hint:

You can use any programming language you choose. An elegant solution will require some planning. I would recommend thinking a bit about the design before you start coding. While you must do you own work, I would recommend comparing final results with others.

Use all your good coding skills: meaningful variable names, print outs of current data, easy subscripting.