CS 357: Algorithmic Game Theory	**Spring 2022**
Assignment 7 (due 04/14/2022)	
Instructor: Shikha Singh	*Solution Template*

Instructions. This is a partner assignment: up to two students can work together and submit a joint write up and receive the same grade. You can invite your partner to jointly edit the same Overleaf document by using the "share" feature. One student in the group should submit the joint PDF on Gradescope, and add the other's name under "Group".

Voting and Social Choice

Problem 1. The *veto rule* is the following social-choice rule:

- Every voter names their least favorite alternative.
- The rule then selects the alternative that is named the least number of times.

(a) Formalize the veto rule as a *positional scoring rule*.[1]

(b) Show that the veto rule is not Condorcet consistent by giving a counterexample.

Problem 2. Is Borda rule Condorcet consistent? Give a proof or a counterexample.

Problem 3. (Independence of Clones Criterion) The *independence of clones* criterion in voting measures a voting mechanism's robustness to strategic nomination. Voting rules, such as plurality, are not robust against strategic nomination because the addition of similar candidates can divide the support among them, which can cause them to lose. We formalize this criterion next.

A set $C \subseteq A$ of alternatives are clones for alternative a if the alternatives in $C \cup \{a\}$ are consecutive in the preference ranking of every voter (they do not need to be in the same order). For example, consider $A = \{A, B, B_2, B_3, C\}$ as the set of alternatives and consider the following preference rankings:

(2 voters) B, B_3, B_2, A, C	(2 voters) B_2, B_3, B, C, A	(3 voters) C, A, B_2, B, B_3

Then, candidates $\{B_2, B_3\}$ are clones of candidate B.

A social-choice function f satisfies *independence of clones* criterion if whenever $f(L) = a$ on a preference ranking L, then $f(L') = a$ (or clone of a) whenever one or more clones are introduced, that is, L' is formed by introducing one of more clones to L.

To see why plurality rule does not satisfy independence of clones criterion, consider the example profiles above with B_2 and B_3 removed. Then, plurality would select B as the winner. However, with the addition of the clones, plurality would select C as the winner.

[1] In lecture, we defined a *positional scoring rule* on $m = |A|$ alternatives as a rule that assigns a score α_j to the alternative ranked in the jth place, such that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$ and $\alpha_1 > \alpha_m$. The rule then elects the alternative with the maximum score.

(a) Does the Borda count satisfy the independence of clones criterion? Give a proof or counterexample.

(b) (**Extra credit/ Optional**) Does ranked-choice voting satisfy the independence of clones criterion? Give a proof or counterexample.

Problem 4. In this question, we consider a voting rule, *Schulze rule* that is a bit complicated to state, but satisfies most of the desirable criteria among preferential voting systems, e.g. Condorcet, Independence of clones, polynomial-time computability, etc.

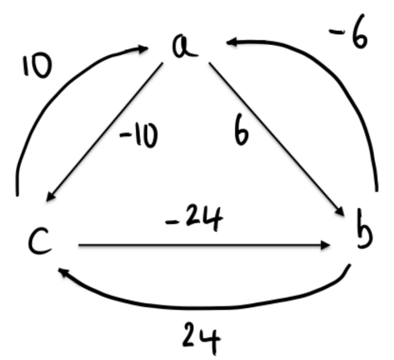
A *weighted-majority graph* is defined as follows: the candidates are the nodes, and there is a directed edge from a to b with weight w_{ab} = (no. of voters who prefer a to b) - (no. of voters who prefer b to a).

The *strength* of a path is defined as the weight of the **least-weight** edge on it. Let $S(x, y)$ be the maximum strength among all paths from x to y.

A candidate a **chain beats** a candidate b if $S(a, b) > S(b, a)$. The Schulze winner is a candidate that chain beats all others (such a winner is surprisingly guaranteed to exist).[2]

Consider an input with three candidates $\{a, b, c\}$, and sixty voters with the following breakdown of ranked orders (on the left), and the corresponding weighted-majority graph (on the right):

23	17	10	8	2
a	b	c	c	b
b	c	a	b	a
c	a	b	a	c



(a) Compute the Schulze winner in the above example.

(b) Show that the Schulze rule is Condorcet consistent.

Problem 5. Consider the greedy strategy to solve the f-manipulation problem, when the social-choice rule f is the Borda rule.

Fix the ranked lists L_{-i} of all other voters. Compute the Borda score s_j of each alternative j under preference lists L_{-i}. Construct the misreport L'_i as follows: place i's favorite candidate a in the top position and rank the other alternatives in ascending order of their Borda scores s_j (that is highest-score candidate goes last).

We say the greedy algorithm is successful if it causes a to win whenever it is possible, given L_{-i}. Prove that the greedy algorithm successfully solves the f-manipulation for Borda rule. (*Hint.* Consider a list L^*_i such that a wins. Show that L^*_i can be transformed to L'_i through a series of swaps, such that a continues to win. This should is similar to how we prove greedy is optimal through exchange argument in CS256.)

[2] Ties can be broken in a consistent way.