

Tideman Voting System - C++ Code Documentation

This documentation provides an overview of the implementation of the Tideman voting system in C++ language. The Tideman algorithm, also known as ranked pairs, is a ranked voting system used to determine the winner in an election based on voter preferences.

Overview

The system works by using voter rankings to calculate preferences between candidates.

It then forms pairs of candidates where one is preferred over the other, sorts these pairs by the strength of the preference, and locks them into a directed graph while avoiding cycles. The winner is the candidate who is not locked in over by any other candidate.

Code Breakdown

Global Variables

- `preferences[MAX][MAX]`: A 2D array to store how many voters prefer one candidate over another.
- `locked[MAX][MAX]`: A 2D array indicating whether a candidate is locked in over another.
- `candidates[MAX]`: An array to store the names of candidates.
- `pairs[MAX * (MAX - 1) / 2]`: An array of pairs where each pair consists of a winner and a loser.
- `pair_count`: Keeps track of the number of pairs.
- `candidate_count`: Stores the number of candidates.

Functions

1. `vote(int rank, string name, int ranks[])`: Updates the ranks based on the vote.
2. `record_preferences(int ranks[])`: Updates the preference matrix after recording the voter's ranks.

3. ``add_pairs()``: Creates pairs of candidates where one is preferred over the other.
4. ``sort_pairs()``: Sorts the pairs based on the strength of the victory.
5. ``lock_pairs()``: Locks the pairs into the graph while ensuring no cycles.
6. ``print_winner()``: Identifies and prints the winner of the election.

Main Function

The main function starts by accepting the candidate names and the number of voters. It then processes the votes, records the preferences, adds pairs, sorts them, locks them into the graph, and finally identifies the winner.

Example Execution

Given the following input:

- Candidates: Alice, Bob, Charlie
- Voters:
 - Voter 1: Alice > Bob > Charlie
 - Voter 2: Bob > Charlie > Alice
 - Voter 3: Bob > Alice > Charlie

The system calculates the preferences, adds and sorts the pairs, locks them into the graph, and prints the winner.

Output:

The winner is: Alice

Conclusion

This C++ implementation of the Tideman voting system allows for efficient handling of ranked-choice votes to determine a winner. The algorithm is implemented using basic C++ constructs, and it ensures that no cycles are created in the pair-locking step to maintain the integrity of the results.