

Voting

Program three different voting strategies

Voting information looks something like:

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

Repopulate the matrix in a variety of ways:

- 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.

Compare three voting methods

- 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.
- 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.
- 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.
- Identify the winner using the Second order Copeland technique.
- 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.