CS 211: Project 2 Discussion

Chris Kauffman

Week 4-2

tally(): Count Votes in a Round

```
tally(CUR_CANDS) : uses field VOTES

error check CUR_CANDS, throw exceptions if needed
create an array of integers TALLIES same size as CUR_CANDS
for every VOTE in the VOTES field
  set CAND to the VOTE's bestChoice( CUR_CANDS )
  set IDX to the index of CAND in CUR_CANDS
  increment element IDX of the TALLIES array

return TALLIES
```

thinHerd(): Eliminate *losers*

```
thin_herd(CUR_CANDS, TALLIES)

error check arguments, throw exceptions if needed
find the MIN and MAX valies in TALLIES

if MIN and MAX are equal
  an all-way tie has occurred, return a copy of CUR_CANDS

set NEXT_CANDS to a copy of CUR_CANDS
for IDX from length of TALLIES-1 to 0
  if the TALLIES[IDX] equals MIN
    remove candidate IDX from NEXT_CANDS
// Q: Why iterate through TALLIES in reverse order?

return NEXT_CANDS
```

tabulate(): Compute a winner with log

```
tabulate() : uses field VOTES

begin the log
set THRESHOLD to the minimum votes needed to win

if validateVotes() returns false
  log that votes are invalid
  end the log and return it

set CUR_CANDS to copy of field CANDIDATES
loop:
  set TALLIES to the results of tally(CUR_CANDS)
  log TALLIES for this round using roundResultsString

  // check for a winner
  loop through all CUR_CANDS
    if TALLIES[i] is greater than THRESHOLD
      log candidate i of CUR_CANDS as the winner
      end the log and return it

  // check for a tie
  if isCompleteTie(TALLIES) returns true
    log the tied candidates   // create a string A, B, C
    end the log and return it

  // eliminate low-vote candidates
  set NEXT_CANDS to results of thinHerd(CUR_CANDS, TALLIES)
  loop through all CUR_CANDS
    if a candidate X from CUR_CANDS is not in NEXT_CANDS
      log that candidate X has been dropped

  set CUR_CANDS to NEXT_CANDS
```