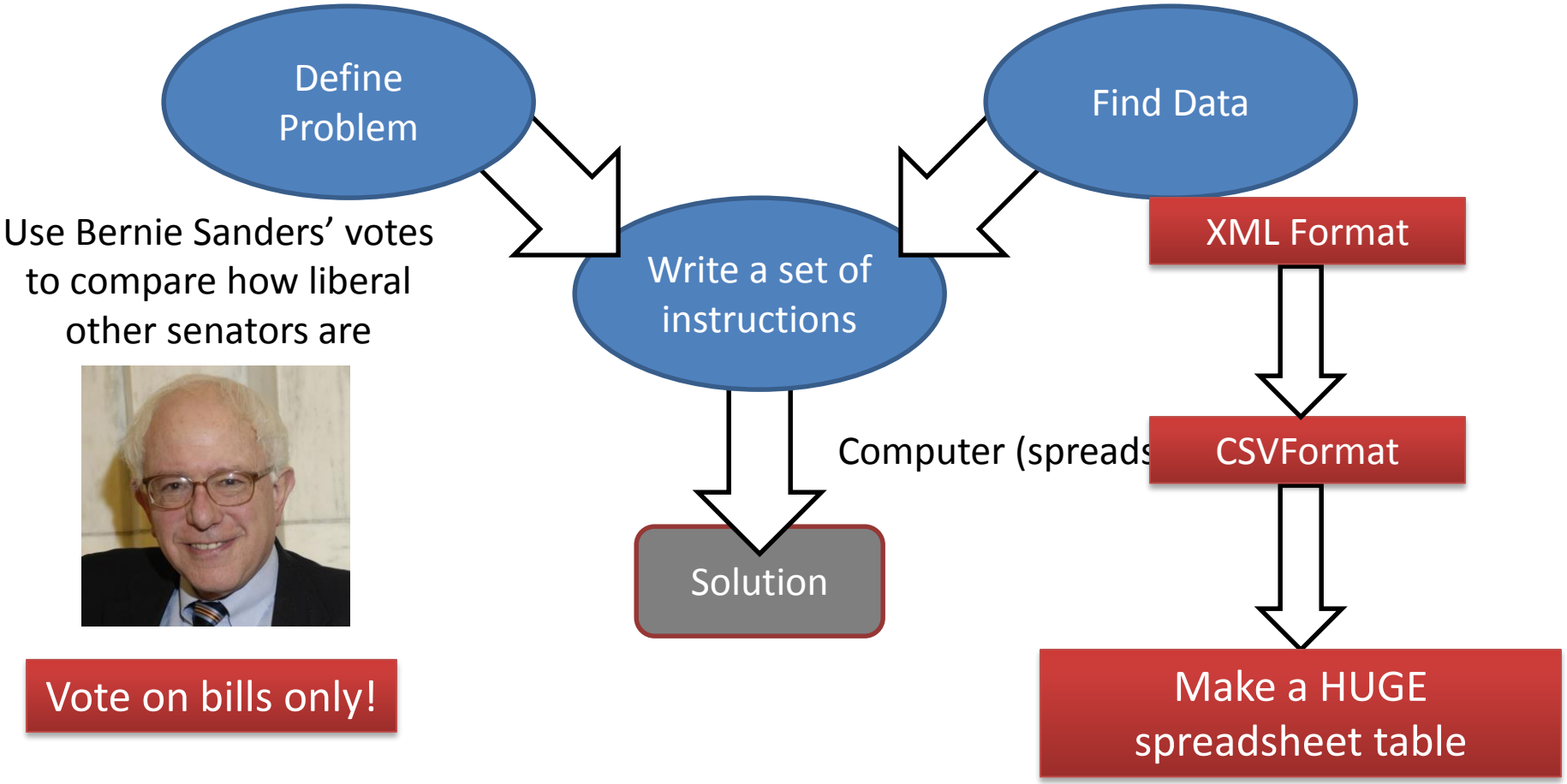
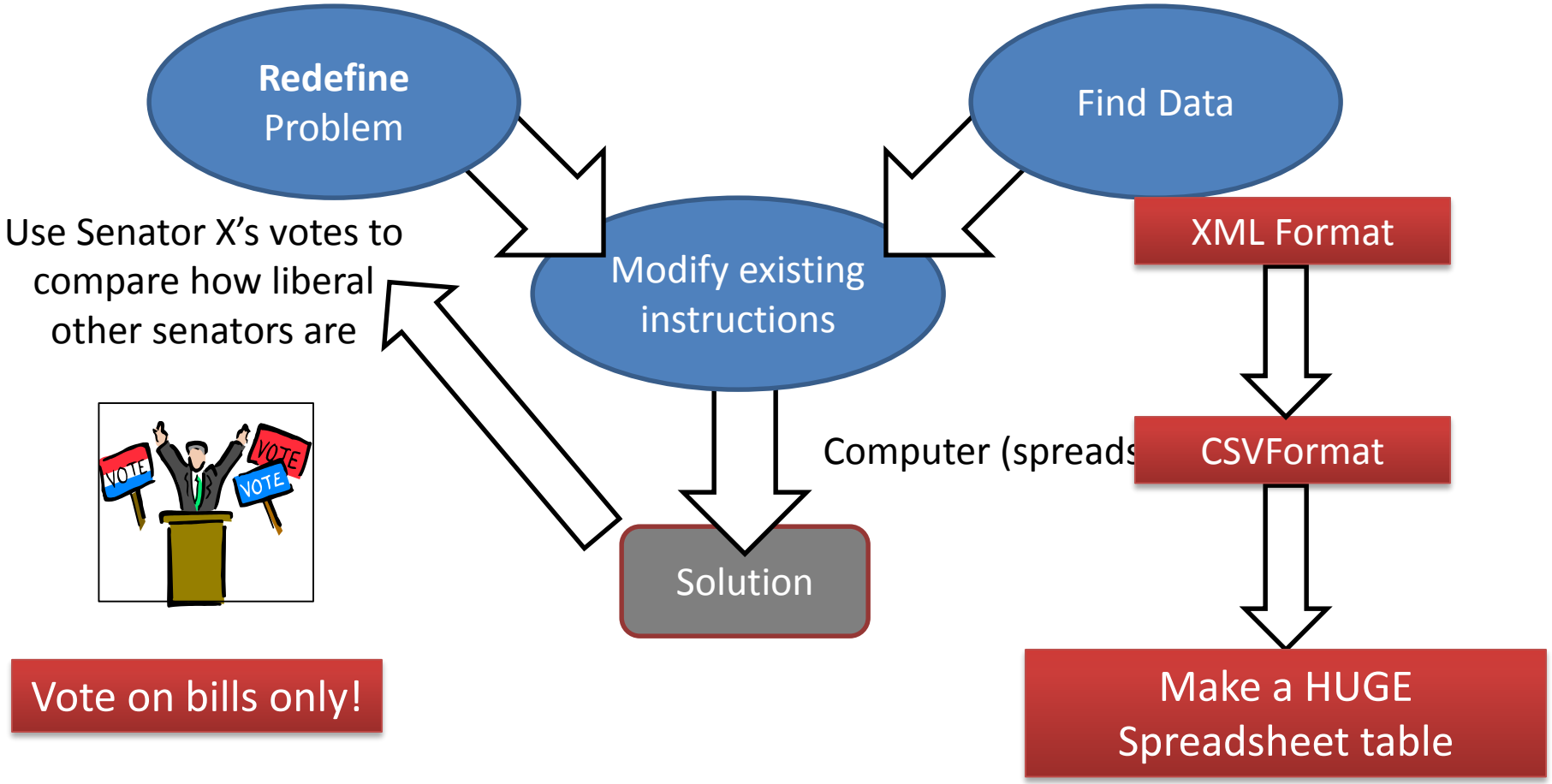
# Ranking Senators with Senator X's Votes

Feb 11, 2016

# What We've Accomplished



# New Problem


Redefine Problem
Find Data
Use Senator X's votes to compare how liberal other senators are
Modify existing instructions
XML Format
Computer (spreads
CSVFormat
VOTE
VOTE
VOTE
Solution
Vote on bills only!
Make a HUGE Spreadsheet table

# In Activity 1-3, you'll...

- Learn new spreadsheet techniques
- Rank relative to any senator, not just Bernie
- Task 2 will make a nice spreadsheet that can be used by others

- Start by doing Task 1 with a partner...

# Reminder about OFFSET

**OFFSET**: Returns a reference to a *range (or cell)* that is a specified number of rows and columns from a base *range (or cell)*.

```
OFFSET(reference, rows, cols)
```

- Go ahead and start on Part 1.

# Let’s do Task 1

# After task 1

- Any questions?
- Move on to Task 2: making a spreadsheet that shows comparison results vs ANY senator, not just Sanders.

# Activity 1-3

- Task 2
  - We broke this task down into manageable pieces.
  - Nice formatting is useful
  - This spreadsheet is now useful for other people
  - It's a *generalization* of the Sanders spreadsheet, and took almost no time to create!
    - If you'd done the Sanders work by hand, redoing it for a new senator would have taken just as long

# Rankings

- Who is least like Bernie?
- What are the rankings relative to that person?
- How do they compare to the Bernie rankings?
  - Rank Bernie on your screen; get a friend to reverse-rank vs Bernie's nemesis. Compare.

# Discussion

- Is ranking relative to Sanders (or Coburn) really a measurement of liberalness? If so, why don't they give exact opposite results?
- What if we had used Warren, or some other very liberal senator, instead of Sanders? Same results? Shuffled?
- Let's look at *all* possible orderings!

# So Far

- What have we accomplished?
  - ACT1-1:
    - Imported voting data from the web
    - Filtered out rows and columns that aren't needed
    - Converted "Yea", "Nay", "Not Voting/Present" into numbers using nested IF statements
    - Made a pivot table to summarize these numbers

# So Far

- What have we accomplished?
  - ACT1-2:
    - Made a new sheet that converts numbers representing **individual votes** into numbers that represent **agreement with Bernie Sanders on each vote**
    - **Absolute addressing** (using the $) was important when we wrote formulas that depended on specific rows
    - Computed a similarity/distance score between each senator and Bernie – we called it **rank**, and it could range from -1 to 1

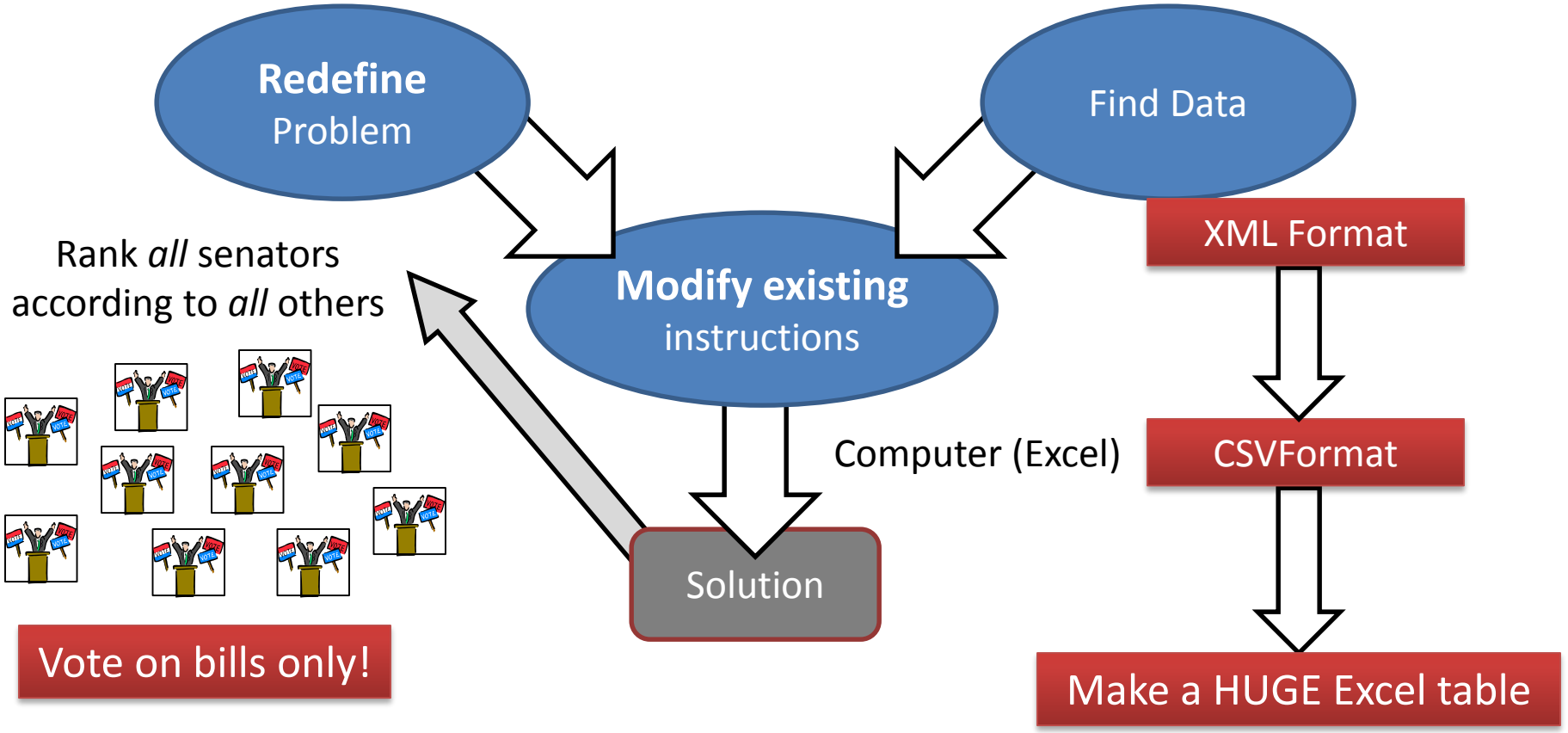
# So Far

- What have we accomplished?
  - ACT1-3:
    - Made a new sheet that converts numbers representing **individual votes** into numbers that represent **agreement with a *selected senator* on each vote**
    - The sheet is interactive; it uses **data validation** to make sure inputs like senator names exist in the table
    - **Healthy use of MATCH and OFFSET** to take the chosen senator name and find the corresponding data in the other sheets

# Next Class

- What if we want to look at how *all* senators rank compared to *all* others?

# Next Class


Redefine
Problem
Find Data
Rank *all* senators
according to *all* others
Modify existing
instructions
XML Format
Computer (Excel)
CSVFormat
Solution
Vote on bills only!
Make a HUGE Excel table

# Compare Every Pair in One Table

What We Have:

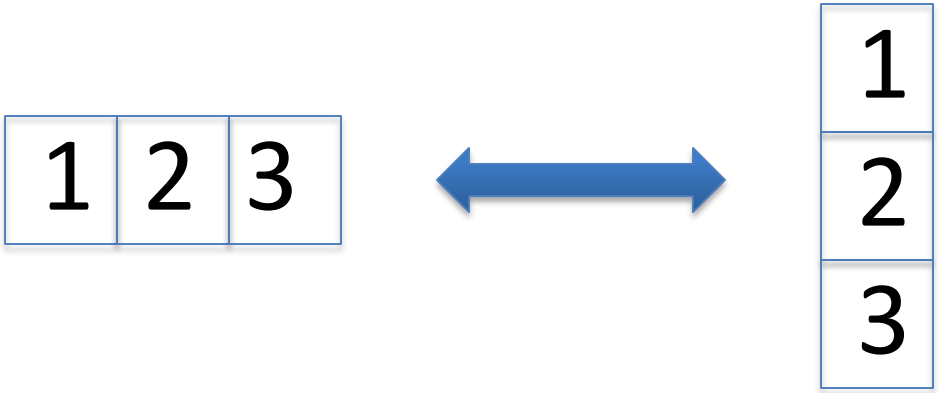
| | 1:101 | 1:115 | 1:129 | 1:138 |
|---|---|---|---|---|
| **Akaka** | Yea | Yea | Yea | Yea |
| **Alexander** | Yea | Yea | Yea | Yea |
| **Ayotte** | Nay | Yea | Yea | Yea |
| **Barrasso** | Yea | Yea | Yea | Yea |

What We Want:

| | Akaka | Alexander | Ayotte | Barraso |
|---|---|---|---|---|
| **Akaka** | 1 | 0.314285714 | 0.085714286 | 0.117647059 |
| **Alexander** | 0.314285714 | 1 | 0.657142857 | 0.647058824 |
| **Ayotte** | 0.085714286 | 0.657142857 | 1 | 0.764705882 |
| **Barrasso** | 0.117647059 | 0.647058824 | 0.764705882 | 1 |

# Next Class

- ACT1-4: Arrays
  - Write formulas where the output is in multiple cells (e.g., *transposing* a row into a column)




  - We'll see how these formulas can help us compute lots of things at once

# Next time

- Build that similarity table
- Following HW, we'll be able to build it with a much simpler method
- How can we make sense of the 10,000 items in it?