# Formulation of A Computational Problem

09/13/2011

# TA hours

- Sunday 7-9 MSLab
- Monday 7-9 MSLab
- Tuesday 7-9 MSLab
- Wednesday 8-10 CIT267

# Today

- Locate data
- Come up with a plan of how to carry out our analysis based on the data
- Learn a foreign language

# Recap

- A senator stands somewhere on the liberal-conservative spectrum
- Hard to decide exactly where; perceptions of which greatly influenced by individual opinions and political interest
- Instead of critiquing opinions, using data helps reveal facts and allows others to verify your theory
- As a first attempt, we want to compare other senators' votes against Ted Kennedy's

# Locating Data

- Where can we find voting records of the congress?
- Look at the data of a particular vote. Did every senator vote? What are the possible votes for a senator?
- Look at the url of the webpage. Do you notice any structure?
- Can you change the url to find the FIRST vote of that congress session?
- And the first vote of the109th congress?

# Now that we have the data…

- The most important thing before you do anything with data, especially large data
- Let's do a back-of-envelope estimation (any guesses beforehand?)
- To do that, we need a step-by-step plan, with each step simple enough so that we know exactly how it's done and how long it takes
- You will do this for almost all projects later on. It helps estimate time; more importantly, it lets you write programs to carry out the plan much, much more easier

# Recipe for solving the problem

- Find out number of votes in 109th congress
- Create a large table, with rows indexed by senator, and columns by votes
- For each vote
  - Open the webpage for that vote
  - If it's on "Passage of a Bill"
    - For each senator
      - Record his/her vote in the appropriate row
- Compare each senator's record with Kennedy's
- Sort the senators by Ted-ness

# How long will it take?

- Find out number of votes in 109th congress
  About 250
- Create a large table, with rows indexed by senator, and columns by votes 100x250 table
- For each vote 250 times …
  - Open the webpage for that vote 10 seconds
  - If it's on "Passage of a Bill" 3 seconds
    - For each senator 100 times …
      - Record his/her vote in the appropriate row/column
        5 seconds

250 x (10 + 3 + 100x5) = 36 hours of work

# How long will it take?

- Compare each senators vote with Kennedy's
  - For each senator 99 times …
    - For each vote 250 times …
      - Record "Y" or "N" according whether the vote matches with Kennedy's 5 seconds
    - Calculate his/her Ted-ness 250 seconds

99 x (250 x 5 + 250) = 41 hours

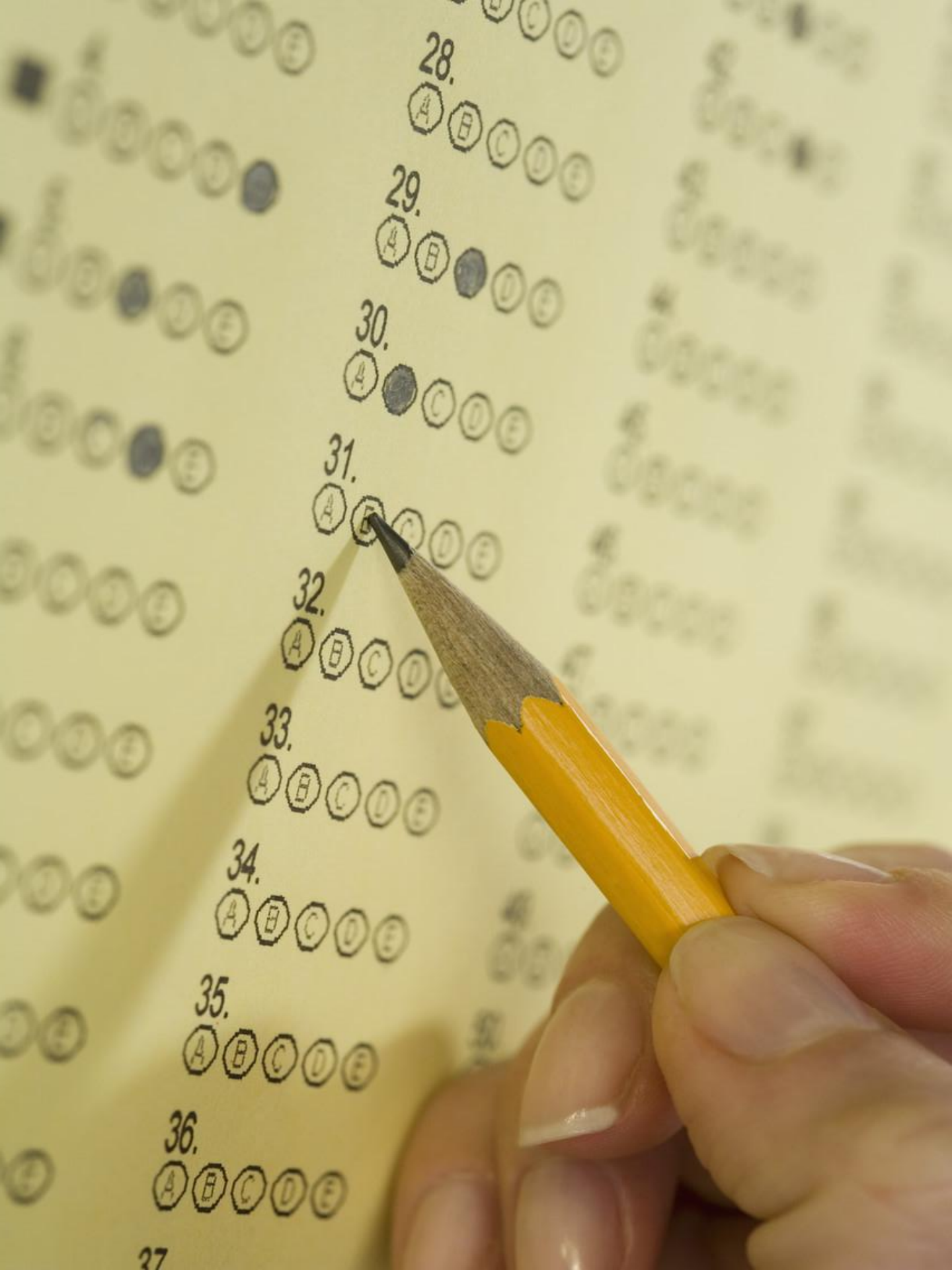
- What if we want to do this comparison for all senators to find the most polarized pair?

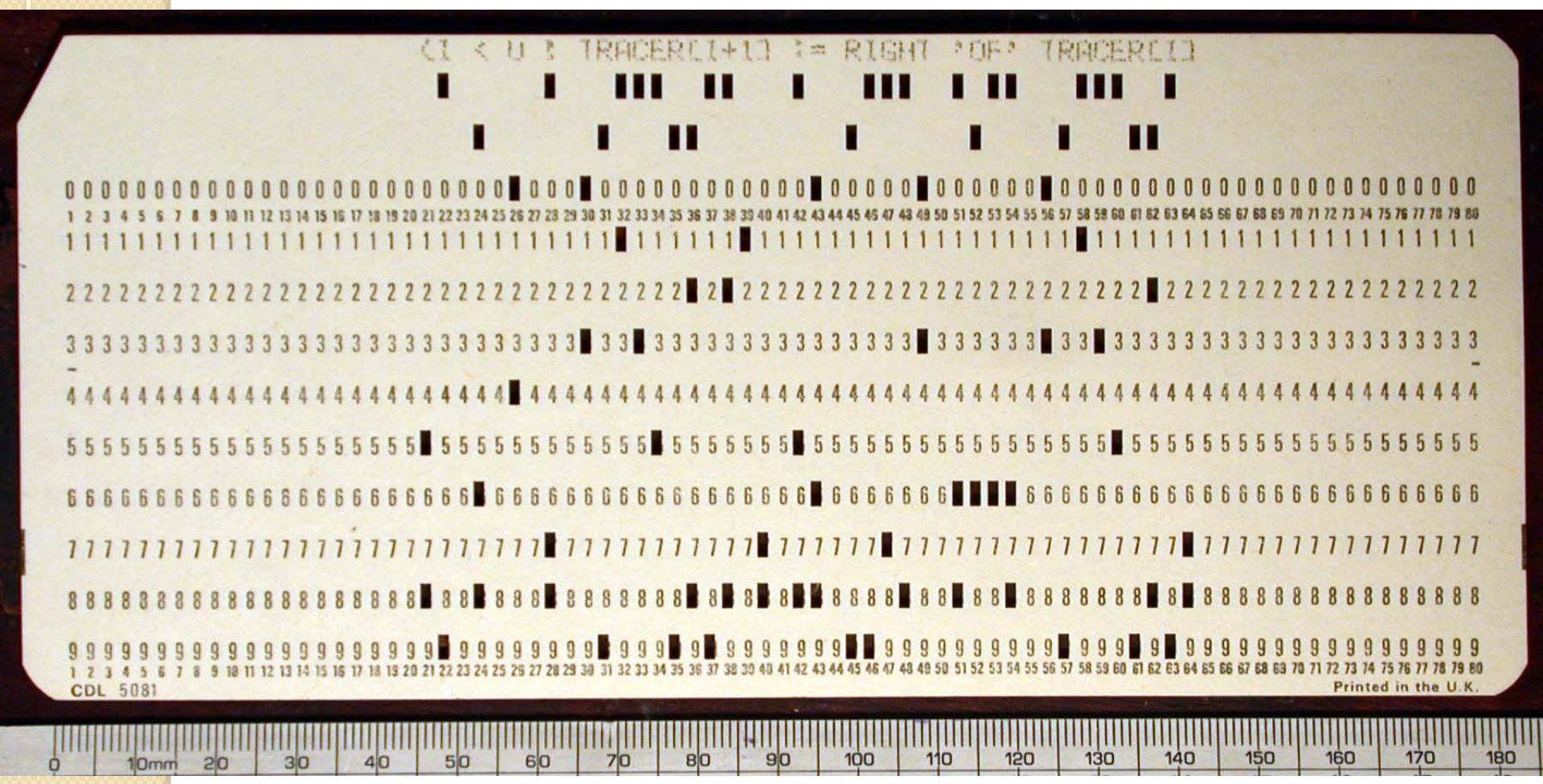
41 x 99 = 4084 hours = 170 days

# Break

# XML

- Extensible Markup Language
- Hard for you to read, but easy for machines to understand; widely used (more on that later)
- Why can't we have both?
- Because humans and machines are good at different things
- Many such examples


28.
29.
30.
31.
32.
33.
34.
35.
36.


CI < U : TRACER[I+1] := RIGHT 'OF' TRACER[I]
CDL 5081
Printed in the U.K.

6 71860 01362 4

```
1. {
2.   "max_id": 27836852555751424,
3.   "results": [
4.     {
5.       "created_at": "Wed, 19 Jan 2011 21:16:37 +0000",
6.       "profile_image_url":
   "http://a2.twimg.com/sticky/default_profile_images/default_profile_1_normal.png",
7.       "from_user_id_str": "191709163",
8.       "id_str": "27836852555751424",
9.       "from_user": "DanLabTesting",
10.      "text": "Twitter api: 1234455",
11.      "to_user_id": null,
12.      "metadata": {
13.        "result_type": "recent"
14.      },
15.      "id": 27836852555751424,
16.      "geo": null,
17.      "from_user_id": 191709163,
18.      "iso_language_code": "en",
19.      "source": "&lt;a href=&quot;http://www.danlabgames.com/index.php?computer=ipad&quot;
   rel=&quot;nofollow&quot;&gt;Wacka Monsta&lt;/a&gt;",
20.      "to_user_id_str": null
21.    },
22.    {
23.      "created_at": "Wed, 19 Jan 2011 21:12:02 +0000",
24.      "profile_image_url":
   "http://a0.twimg.com/profile_images/1142619698/DSC_0195_normal.jpg",
25.      "from_user_id_str": "165544885",
26.      "id_str": "27835698383945728",
27.      "from_user": "Deberamatkin",
28.      "text": "Fetching the number of followers without using any Twitter API
   http://pr9.in/4q",
29.      "to_user_id": null,
30.      "metadata": {
31.        "result_type": "recent"
32.      },
33.      "id": 27835698383945728,
34.      "geo": null,
35.      "from_user_id": 165544885,
36.      "iso_language_code": "en",
```

# XML's brother HTML

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<roll_call_vote>
    <congress>107</congress>
    <session>1</session>
    …
    <document>
        <document_type>H.R.</document_type>
        <document_number>333</document_number>
        …
    </document>
    <members>
        <member>
            <member_full>Akaka (D-HI)</member_full>
            <last_name>Akaka</last_name>
            <party>D</party>
            <vote_cast>Yea</vote_cast>
            …
            </member>
        <member>
        …
        </member>
    </members>
</roll_call_vote>
```

<?xml version="1.0" encoding="UTF-8" ?>

- *"We're using XML; the character set we're using is a really common one"*
- Stuff in pointy brackets <…> describes the document
- Stuff outside is the *content*

<roll_call_vote>

…

</roll_call_vote>

- Almost all brackets contain *tags*
- They come in matching pairs
  - <foobar> … </foobar>
- Names are generally human-readable
- Names become column-labels in Excel!

# Now you may wonder…

- Why do we want to learn about this?
  - Yeah, we want to look at this voting data, and it happens to be in this format, and Excel happens to be able to read it… but, I mean, the data could be in any other forms, and I don't really care…
- Because pretty much everything is XML, or like XML…







# Proof: Microsoft Word is XML

- On your desktop, right click and select "New" → "Microsoft Word Document"
- Rename, edit and save (write your favorite quote, make it italic and red, etc)
- Right click, select "Open with …" Then click on "other programs", and choose "Notepad"
- What do you see?

# Not quite working…

- Because the file is "zipped"
- Have you ever unzip a file?
- To make Windows recognize a zipped file, you need to change its "extension": change the name to "xxxx.docx.zip"
- Right click, and select "Extract to xxxx.docx\"
- Open that folder, and behold!

# A lot of stuff in here…

- Most of them deals with versions, authors, time and file infrastructures
- Look in the folder "word", use Internet Explorer to open "document.xml"
- Find the text you entered, and try to make some sense out of the whole mess