# Plan for TBT

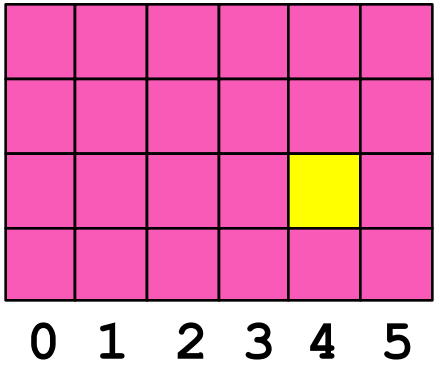
- **Review APTs**
  - Sorting with itemgetter
  - Nested Loops

- **Review Recommender Assignment**
  - How do you get recommendations
    - Based on Yelp, Amazon, Netflix,
  - What is collaborative filtering
  - How to get assignment done

# Patterns in some APTs

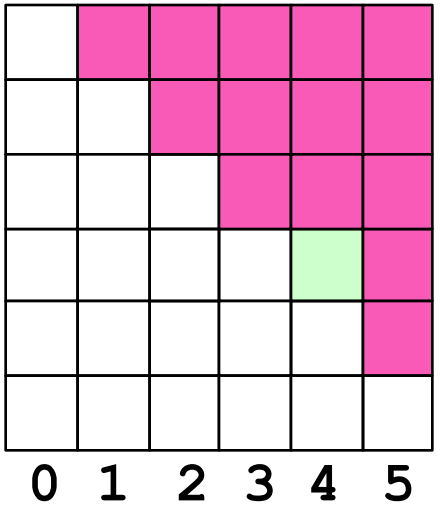
- **PositiveID and FriendScore APT (challenge)**
  - Nested loops, look at everything: look at pairs
  - While True: … break

```
for i in range(4):
    for j in range(6):
        elt = m[i][j]
```




m[2][4]

```
for i in range(6):
    for j in range(i+1,6):
        elt = m[i][j]
```




m[3][4]

# How do we store/use tabular data?

- How can you consider all pairs, e.g., PositiveID?
  - Do suspects 2 and 5 share same characteristics as 5 and 2?
    - Which loop on previous slide captures this?
- What is a matrix?
  - Table of numbers? Entries? Two-Dimensional, m X n
  - Many arithmetic operations available
  - Convenient for representing correlated data
- In Python what is a list of lists?
  - Other languages have arrays, matrixes, …

# Sorting review

- Using operator.itemgetter:
  - Why use reverse=True?
- What does a stable sort get us?
  - Sort by tie-breaking criteria first, then resort

```
sorted([tup,key=operator.itemgetter(1)])
sorted([tup,key=operator.itemgetter(1,0)])
```

# Questions

http://bit.ly/101fall15-nov24-1

# John Tukey: 1915-2000

- Cooley-Tukey FFT
- Bit: Binary Digit
- Box-plot
- "software" used in print

*Far better an approximate answer to the* **right** *question, which is often vague, than an* **exact** *answer to the wrong question, which can always be made precise.*

*The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.*

# Math, Engineering, Sociology

- **Netflix prize in 2009**
  - **Beat the system, win**
  - **http://nyti.ms/sPvR**


NETFLIX

# Collaborative Filtering

- How does Amazon know what I want?
  - Lots of customers, lots of purchases
- How does Pandora know music like Kanye's?
  - This isn't really collaborative filtering, more content-based
- How doe Netflix recommend movies?
  - Why did they offer one million $$ to better their method?
- Students at Duke who like Compsci also like …
  - Could this system be built?

# From User Rating to Recommendations


| Spectre | Martian | Southpaw | Everest | PitchPerfect 2 |
|---|---|---|---|---|
| 3 | -3 | 5 | -2 | -3 |
| 2 | 2 | 3 | 2 | 3 |
| 4 | 4 | -2 | 1 | -1 |

- **What should I choose to see?**
  - What does this depend on?
- **Who is most like me?**
  - How do we figure this out

# Data For Recommender

- **Users/Raters rate Items**
  - We need to know the items
  - We need to know how users rate each item
- **Which eatery has highest average rating?**
  - Conceptually: average columns in table
  - How is data provided in this assignment?

| | ABP | BlueEx | McDon | Loop | Panda | Nasher |
|---|---|---|---|---|---|---|
| Sam | 0 | 3 | 5 | 0 | -3 | 5 |
| Chris | 1 | 1 | 0 | 3 | 0 | -3 |
| Nat | -3 | 3 | 3 | 5 | 1 | -1 |

# Data For Recommender

- Items are provided in a list of strings
  - Parsing data provides this list
- Ratings provided in dictionary
  - Key is user ID
  - Value is list of integer ratings

| | ABP | BlueEx | McDon | Loop | Panda | Nasher |
|---|---|---|---|---|---|---|
| Sam | 0 | 3 | 5 | 0 | -3 | 5 |
| Chris | 1 | 1 | 0 | 3 | 0 | -3 |
| Nat | -3 | 3 | 3 | 5 | 1 | -1 |

# Data For Recommender

- **Given Parameters**
  - items: a list of strings
  - ratings: dictionary of ID to ratings list
- **Can you write**
  - getAverage("ABP", items, ratings)

| | ABP | BlueEx | McDon | Loop | Panda | Nasher |
|---|---|---|---|---|---|---|
| Sam | 0 | 3 | 5 | 0 | -3 | 5 |
| Chris | 1 | 1 | 0 | 3 | 0 | -3 |
| Nat | -3 | 3 | 3 | 5 | 1 | -1 |

# Drawbacks of Item Averaging

- Are all ratings the same to me?
  - Shouldn't I value ratings of people "near" me as more meaningful than those "far" from me?
- Collaborative Filtering
  - https://en.wikipedia.org/wiki/Collaborative_filtering
  - How do we determine who is "near" me?

- Mathematically: treat ratings as vectors in an N-dimensional space, N = # ratings
  - Informally: assign numbers, higher the number, closer to me

# Collaborative Filtering: Recommender

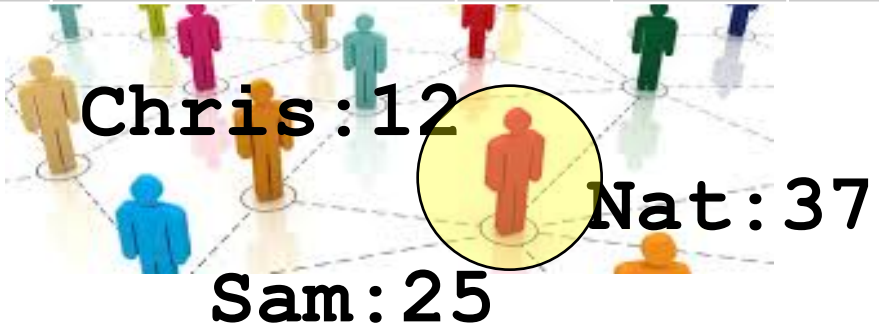
- First determine closeness of all users to me:
  - "Me" is a user-ID, parameter to function
  - Return list of (ID, closeness-#) tuples, sorted

- Use just the ratings of person closest to me
  - Is this a good idea?
  - What about the 10 closet people to me?
- What about weighting ratings
  - Closer to me, more weight given to rating

# Collaborative Filtering

- For Chris: `12 * [1,1,0,3,0,-3] =`
  - `[12,12,0,36,0,-36]`
- For Sam: `[0,75,125,0,-75,125]`

| | ABP | BlueEx | McDon | Loop | Panda | Nasher |
|---|---|---|---|---|---|---|
| Sam | 0 | 3 | 5 | 0 | -3 | 5 |
| Chris | 1 | 1 | 0 | 3 | 0 | -3 |
| Nat | -3 | 3 | 3 | 5 | 1 | -1 |



# Adding lists of numbers

```
[12, 12,   0, 36,  0,-36]
[ 0, 75, 125,  0,-75,125]
[35, 25, -25, 15,  0, 50]
--------------------------
[47, 112,100, 51,-75,139]
```

- Adding columns in lists of numbers
  - Using indexes 0, 1, 2, ... sum elements of list
  - `sum([val[i] for val in d.values()])`

# Processing Data for Recommender

- **Data comes in many formats, but all formats must produce:**
  - **List of items being rated**
  - **Dictionary of raterID/names to list of ratings**
- **Read/Parse data files: create list, dictionary**
  - **Return JSON strings, create list/dict from these**

```
[ABP,Blue Express,Washington Duke]
```

```
{Chris: [1,1,0,3,0,-3],
 Nat: [-3,3,3,5,1,-1] }
```

# JSON format

- Using the json library
  - json.dumps(structure) creates a string st
  - json.loads(st) recreates the structure

- Allows transmission of structured data over the Internet, for example
  - Standard, JavaScript Object Notation (JSON)
  - Python dictionaries are in JSON format!

# Reader modules: Food Format

- All Reader modules return a tuple of strings: item list and ratings dictionary
  - Provided!

```
Nasher Cafe
ABP
WaDuke
Loop
Panda
Penn Pavilion
McDonalds
Blue Express
student1001,3,3,-3,5,3,1,0,0
student1002,0,1,5,-3,1,0,5,-3
student1003,5,-3,3,0,3,0,3,-3
student1004,0,1,0,0,1,0,5,0
```

# Reader modules: Book Format

- All Reader modules return a tuple of strings: item list and ratings dictionary
  - Alternate form of data, all information on line line of comma-separated values

```
student1001,Nasher,3,ABP,3,WaDuke,-3,Loop,5
student1002,Nasher,0,ABP,1,WaDuke,5,Loop,-3
```

# Reader modules: Movie Format

- All Reader modules return a tuple of strings: item list and ratings dictionary
  - Alternate form of data, each rating on one line, lines intermixed, no zeros stored
  - Advantages? Disadvantages?

```
student1001,Nasher,3,
student1002,ABP,1
student1001,ABP,3,
student1002,WaDuke,5
```