Assignment 3 Discussion

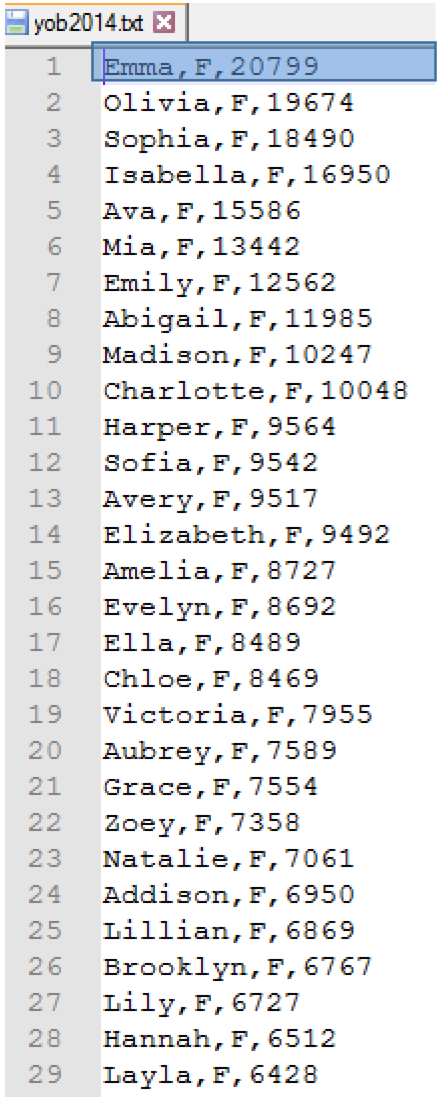
Due Date: Mar 9, Thu, 10 pm

General Goals

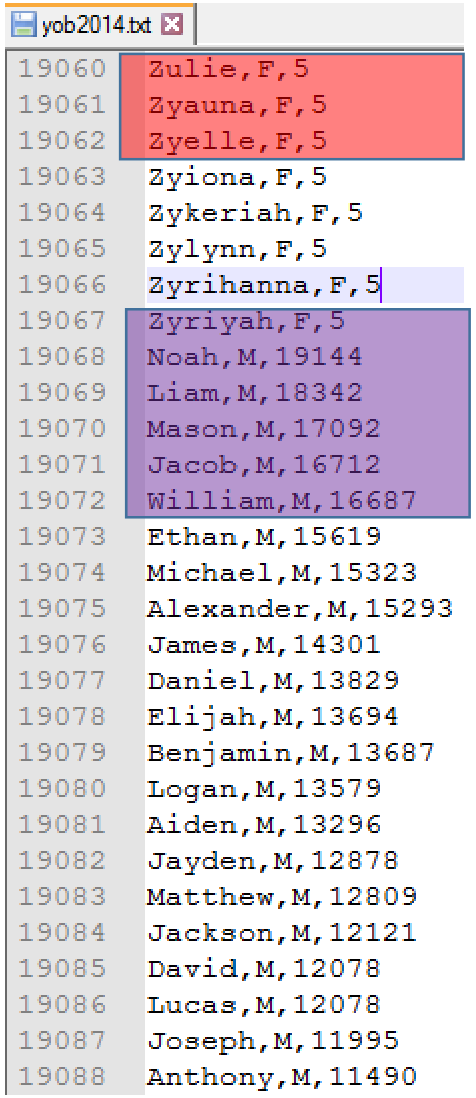
- Tree Representation
  - Read data about tree nodes and construct a tree based on that.
  - Compare the performance of Tree data structure with other well-known data structures
    - HashMap
    - Array /Linked List
  - Binary Search Tree and AVL

Pair Team Programming

Specific Goals

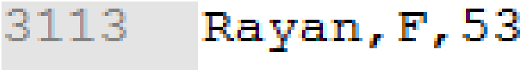


```
 1  Emma,F,20799
 2  Olivia,F,19674
 3  Sophia,F,18490
 4  Isabella,F,16950
 5  Ava,F,15586
 6  Mia,F,13442
 7  Emily,F,12562
 8  Abigail,F,11985
 9  Madison,F,10247
10  Charlotte,F,10048
11  Harper,F,9564
12  Sofia,F,9542
13  Avery,F,9517
14  Elizabeth,F,9492
15  Amelia,F,8727
16  Evelyn,F,8692
17  Ella,F,8489
18  Chloe,F,8469
19  Victoria,F,7955
20  Aubrey,F,7589
21  Grace,F,7554
22  Zoey,F,7358
23  Natalie,F,7061
24  Addison,F,6950
25  Lillian,F,6869
26  Brooklyn,F,6767
27  Lily,F,6727
28  Hannah,F,6512
29  Layla,F,6428
```

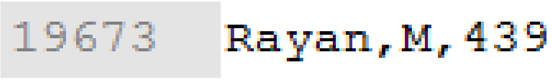


```
19060  Zulie,F,5
19061  Zyauna,F,5
19062  Zyelle,F,5
19063  Zyiona,F,5
19064  Zykeriah,F,5
19065  Zylynn,F,5
19066  Zyrihanna,F,5
19067  Zyriyah,F,5
19068  Noah,M,19144
19069  Liam,M,18342
19070  Mason,M,17092
19071  Jacob,M,16712
19072  William,M,16687
19073  Ethan,M,15619
19074  Michael,M,15323
19075  Alexander,M,15293
19076  James,M,14301
19077  Daniel,M,13829
19078  Elijah,M,13694
19079  Benjamin,M,13687
19080  Logan,M,13579
19081  Aiden,M,13296
19082  Jayden,M,12878
19083  Matthew,M,12809
19084  Jackson,M,12121
19085  David,M,12078
19086  Lucas,M,12078
19087  Joseph,M,11995
19088  Anthony,M,11490
```

- Write a program that analyzes a text file (comma-delimited) containing baby names
- Format of each record:
  - name,gender,number
  - name: Baby's name (2 to 15 characters)
  - Gender: Baby's gender (M or F)
  - Number: Number of occurrences of the name.
  - File is sorted
    - first on gender
    - Then on number of occurrences in descending order.
    - Tie on the number of occurrences: Names are listed in alphabetical order.
- Some names can be for male and female
  - Example: Rayan



```
3113  Rayan,F,53
```



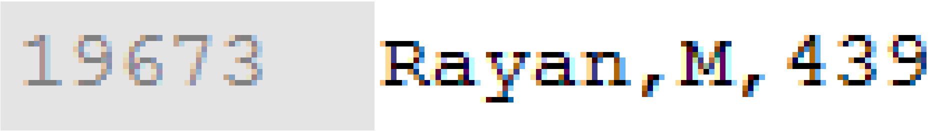
```
19673  Rayan,M,439
```

What to do

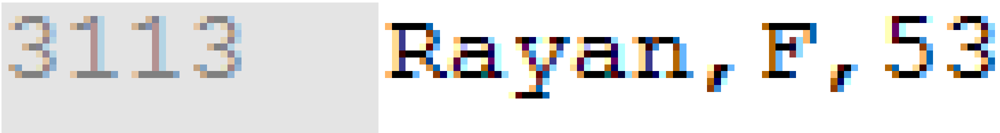
- Your program must store the names and the counts of the names in
  - A single binary search tree
  - Hashmap
  - Array/Linked list.
- Each name can only be stored once using the given data structure.
- **You must not use Java's build-in tree class and create your own tree class.**

What to do

- Your program must contain the following methods:
  - **SearchName:**
    - For a name returns number of male and female babies born in that year who has that name.
    - Return a rank for this name (how popular is this name for boys and girls).
  - Example
    - Selected Name: Rayan



```
19673   Rayan,M,439
```



```
3113   Rayan,F,53
```

- Output:

```
Year     Male     Rank-Male     Female     Rank-Female
2014     439      606           53         3113
```

What to do

- Your program must contain the following methods:
  - **MostPopularName**:
    - Returns the most popular 10 male and female baby names for a given year with their numbers and percentage of babies with that name.
  - Output Example

Female Name	Frequency	%	Male Name	Frequency	%
Emma	20799	1.1758986	Noah	19144	1.00685
Olivia	19674	1.1122952	Liam	18342	0.96467
Sophia	18490	1.0453562	Mason	17092	0.898928
Isabella	16950	0.9582903	Jacob	16712	0.878942
Ava	15586	0.8811748	William	16687	0.877628
Mia	13442	0.759961	Ethan	15619	0.821458
Emily	12562	0.710209	Michael	15323	0.80589
Abigail	11985	0.6775876	Alexander	15293	0.804312
Madison	10247	0.5793275	James	14301	0.75214
Charlotte	10048	0.5680768	Daniel	13829	0.727315

What to do

- Your program must contain the following methods:
  - **ShowNameAlphabetically**
    - Display names in alphabetical order
    - Next to each name prints the number of male and female babies that have this name and percentage of babies (male and female) for that name.

Name	Gender	Frequency	%
Aaban	M	16	0.000841496
Aabha	F	9	0.000508827
Aabriella	F	5	0.000282682
Aadam	M	19	0.000999276
Aadan	M	8	0.000420748
Aadarsh	M	18	0.000946683
Aaden	M	236	0.012412064
Aadhav	M	25	0.001314837
Aadhi	M	5	0.000262967
Aadhira	F	13	0.000734972
Aadhya	F	249	0.01407754
Aadi	M	54	0.002840048
Aadian	M	5	0.000262967
Aadil	M	11	0.000578528
Aadit	M	31	0.001630398
Aadith	M	9	0.000473341
Aaditri	F	10	0.000565363
Aaditya	M	40	0.00210374
Aadiv	M	5	0.000262967

…………………………………………………………………………………………..

…………………………………………………………………………………………

Grading

- What to submit on Canvas
  - All files that are needed to compile and run your program.
  - Your write-up.
  - Your test files (at least two different test files)

Component	**Points**
SearchName	15
MostPopularName	15
ShowNameAlphabetically	15
Write-up and Test Cases	6
Contribution Summary	4

Grading: Write-Up & Test Cases (5 points)

- A write-up of your implementation.
- Plain text file (writeup.txt) which should have
  1. Your names
  2. Among the three data structures you used, which one you think perform the best. Why?
  3. Any parts of your program doesn’t work? Discuss.
     - Failure to disclose obvious problems will result in additional penalties.
  4. Any parts of your program is inefficient? Discuss
  5. What parts of the assignment were most challenging? Discuss
  6. A discussion on how you approached testing that your program was correct and asymptotically efficient.

Grading: Contribution Summary (2 points)

- Submit via email to me and TA **(Not in canvas)**
- Every student submit separately
- What to submit
  1. Your names
  2. One paragraph:
     - Explaining your contribution.
     - Include examples of methods you implemented, how you contributed to the design, bug fixing efforts, etc.
  3. One paragraph:
     - Explain your partners' contribution
     - Include examples of methods your partner implemented, design ideas, bug fixes, etc.
  4. Any collaboration problem experienced? How you approached the problem and solved it?

How the program works

- Create three data structures (Tree, Hash map, and Array/Linked List)
- Read the txt file (yob2014.txt) and load the data to three data structures
- Loop repeatedly
    - Ask user for preferred data structure
        - 1 = tree, 2 = hash map, and 3= linear data structure
    - Ask user which information s/he wants to know
        - 1 = Search for a name, 2 = Most-Popular Name, and 3 = Show Name Alphabetically
        - For option 1 (Search for a name)
            - Ask user to enter a name.
                - If there is a match, display name and number of male and female babies sharing that name.
                - If no match, a descriptive message should be printed (i.e., this name does not exist in the name database) and the program would stop.
        - For option 2 (Most-Popular Name)
            - Display the most popular 10 male and female baby names for a given year with their numbers and percentage of babies with that name.
        - For option 3 (Show Name Alphabetically)
            - Display the names in alphabetical order
            - Next to each name prints the number of male and female babies that have this name and percentage of babies (male and female) for that name.
    - Ask user if s/he wants to start over.
        - If yes, continue the loop
        - If not then exit from the loop and also exit from the program