Chapter 8, Strings:

- General introduction; string sorting (last week Friday)
- Tries (**Today**)
- Other string topics (Wednesday)
  - Regular expressions
  - ~~Huffman encoding~~
  - ~~Edit distance~~
  - ~~Grammars and parsing~~
- Review for Tests 3 and 4 (next week Friday)

Today:

- Problem statement
- Main idea behind tries
- Code details:
  - Node class
  - Find
  - Insertion
  - Deletion

Coming up (the last):

Catch up on old projects . . .
Do **Perfect Hashing** *project (due today, Monday, Apr 22)*
Do **Trie** *project (due Friday, Apr 26)*

Due **Today, Mon, Apr 22**
Read Section 8.2

Projects:

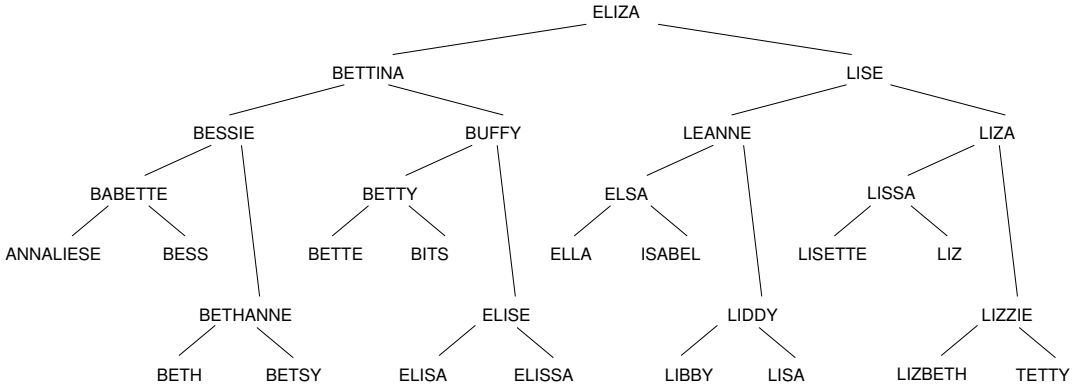
- Last regular project score update on Tues, Apr 23
- "Two minute warning" run of scripts on Fri, Apr 26 (*no Cancas update—see report file* `/cslab/class/cs345/(your userid)/(your userid).report`)
- All projects due on the last day of *classes*, **midnight between Fri, Apr 26 and Sat, Apr 28**—*not last day of finals*.

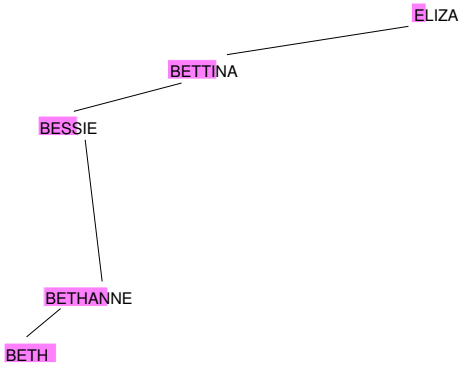
Final exam

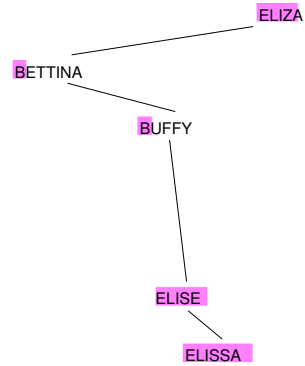
- Our final exam block is **Tues, Apr 30, 10:30am–12:30pm**
  The first time ever that I have given a final exam in April.
- During our final exam block, we will meet in the **CSCI lab**
- Test 3 ("written"/conceptual part) will be like Test 1, but covering BSTs (ch 5) through strings (ch 8)
- Test 4 (programming part) will work the same way as Test 2, covering dynamic programming, hashing, and strings.

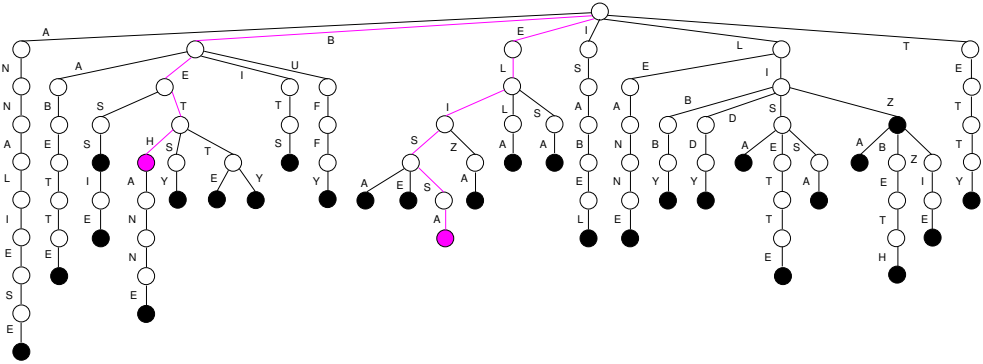
In class and in the text, we see an iterative implementation of trees.

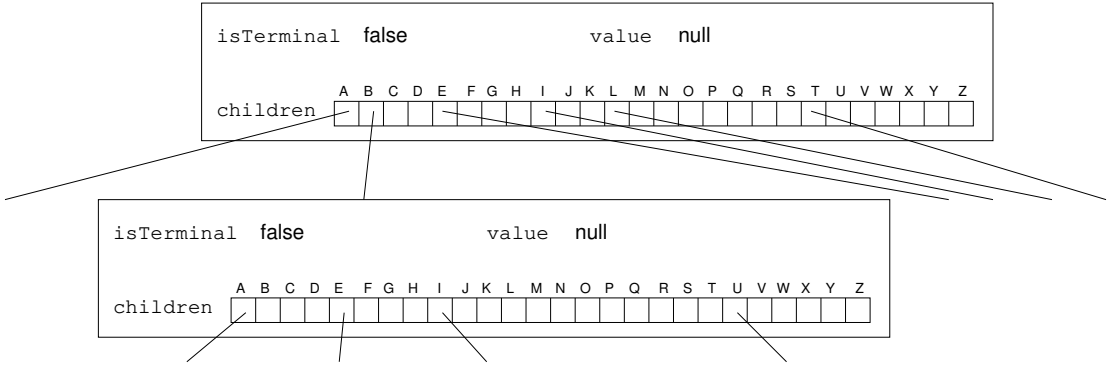
In the accompanying project, you'll implement the operations recursively in the node.


ELIZA
BETTINA
LISE
BESSIE
BUFFY
LEANNE
LIZA
BABETTE
BETTY
ELSA
LISSA
ANNALIESE
BESS
BETTE
BITS
ELLA
ISABEL
LISETTE
LIZ
BETHANNE
ELISE
LIDDY
LIZZIE
BETH
BETSY
ELISA
ELISSA
LIBBY
LISA
LIZBETH
TETTY


ELIZA
BETTINA
BESSIE
BETHANNE
BETH


ELIZA
BETTINA
BUFFY
ELISE
ELISSA


A
B
E
I
L
T
N
A
E
I
U
L
S
E
I
E
N
B
S
T
T
F
I
L
S
A
A
B
D
S
Z
T
A
E
S
H
S
T
S
F
S
Z
A
A
B
N
B
D
A
E
S
A
B
Z
T
L
T
I
A
Y
E
Y
Y
A
E
S
A
E
N
Y
Y
T
A
E
I
Y
I
T
E
N
A
L
E
T
T
E
E
E
E
E
H
S
E
E


isTerminal false
value null
children
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
isTerminal false
value null
children
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Invariant 39. [Class invariant of `TrieMap`]

(a) For all nodes, the path to that node is a prefix to at least one key in the map.

(b) For all nodes, the node is terminal iff the path to that node is a key in the map.

In class and in the text, we see an iterative implementation of trees.

In the accompanying project, you'll implement the operations recursively in the node.