CSE 100 Lecture 21
Tries
Announcements

• PA4 writeup is now available
• Skeleton code will be released on Wed.
• Please read the writeup thoroughly (also the focus of today’s discussion)
• Challenge: SETUP and DESIGN (not the coding per se) is harder than previous PAs
• Before Thanksgiving:
  ‣ Choose your partner, start thinking about PA4, start discussing!
This is the boggle board… how would you represent it in your code?
A. As a linked list
B. As a 2D array
C. As a tree
D. Graph
Graphs in PA4 (HINT!!)

This is also a graph... where are the edges? (fill them in)
Graphs in PA4 (HINT!!)

So if you use a 2D array to represent the board, how do you store the nodes? How do you store the edges?
What graph representation will you use?
A. An adjacency list
B. An adjacency matrix
C. 2D array
D. B and C are the same (either one works)
Two searches in PA4

Board

S T N G
E I A E
D R L S
S E P O

Dictionary
race
radiant
rain
ran
rat
rats
...

race
radiant
rain
ran
rat
rats
...
Two searches in PA4

Parallel search allows you to minimize time by ruling out unneeded paths. But how to represent the tree.
Tries: Efficient way to store/find keys that are sequences of digits

Create the code trie for the following code:

A: 00001
S: 10011
E: 00101
R: 10010
C: 0001
H: 101
Does the structure of the trie depend on the order in which keys are inserted?
A. Yes  B. No
Multi-way tries: Efficient finding of keys by their sequence

Build the trie which holds the following number keys:

8
1234
59
123
8775
80

Assuming your trie could potentially hold any decimal number, how many children does each node (potentially) have?
A. 2    B. 8    C. 10    D. Other
Multi-way tries: Efficient finding of keys by their sequence

Build the trie to store the following numbers:

8
1234
59
123
8775
80

Which node in the trie represents 1234?

A. Red
B. Purple
C. Blue
D. Green
E. Other
Multi-way tries: Efficient finding of keys by their sequence

Build the trie to store the following numbers:

8
1234
59
123
8775
80

Is there a way to find whether all keys contained in a sequence of digits are present in the trie?

A. Yes
B. No
Properties of tries

Build the trie to store the following numbers:

\begin{itemize}
  \item 8
  \item 1234
  \item 59
  \item 123
  \item 8775
  \item 80
\end{itemize}

Is there a “strong ordering” property in a trie? That is, are smaller keys always to the left of larger keys?
A. Yes \hspace{1cm} B. No
Suppose your keys are a sequence of at most $D$ digits, and $N$ is the maximum number of keys you will store. What is the worst case height of this trie (i.e., for large $N$)?

A. $D$  B. $\lg(D)$  C. $N$  D. $D \times N$  E. Other
Properties of tries

Build the trie to store the following numbers:

If you stored the same N D-digit keys in a Binary Search Tree, what would be the worst case height of the tree?

A. N  
B. \( \lg(10^D) \)  
C. \( \lg(N) \)  
D. \( \lg(D) \)  
E. Other
Consider storing the full $10^D$ keys. We know that on average the height of a BST will be $\log(10^D)$. Which is smaller: $D$ or $\log(10^D)$?

A. $D$  
B. $\log(10^D)$  
C. They are the same
Properties of tries

Build the trie to store the following numbers:

8
1234
59
123
8775
80

So what is the main drawback of tries?
A. They are difficult to implement
B. They (usually) waste a lot of space
C. They are slow
D. There is no drawback of tries
Ternary search trees to the rescue!

- Tries combine binary search trees with tries.
- Each node contains the following:
  - A key `digit` for search comparison
  - Three pointers:
    - `left` and `right`: for when the digit being considered is less than and greater than (respectively) the digit stored in the node (the BST part)
    - `middle`: for when the digit being considered is equal to the digit stored in the node (the trie part)
  - An `end` bit to indicate we’ve completed a key stored in the tree.
Draw the ternary tree for the following (in this order)

i
just
met
this
is
crazy
call
me
maybe
List all the words (strings) you can find in this TST

Are the following in the tree? (A=yes, B=no)
• get
• if
• gif
• its
• gacar
• tsem
Draw the ternary tree for the following (in this order)

i
just
met
this
is
crazy
call
me
maybe

Does the structure of the tree depend on the order in which keys were inserted?  A. Yes   B. No
Draw the ternary tree for the following (in this order)

i
just
met
this
is
crazy
call
me
maybe

Does the tree have a strong ordering property (i.e. keys in the left subtree are always less than trees in the right subtree? 
A. Yes B. No
Algorithms for insert and find (in TSTs and MWTs)

- In your reading and/or in Paul Kube’s slides
- In discussion next week