StateTree.java

Objective: To implement a binary tree and perform operations on that tree.

Background:
We have written a linked list, a simple data structure useful in some applications. Unfortunately, it is very inefficient when we need to find data. Average case, we need to traverse half of the list before we find what we want, an O(n) algorithm. This would bog down any search engine that needs information quickly.

A better solution is to create a database that can be accessed using a "divide and conquer" method like we used in Merge Sort. The Binary Tree is one solution. The tree structure stores data in such a way that each comparison with an element reduces the number of elements remaining to search in half. Average case, a search will take O(log n) time. This is a far better database for searching than the linked list.

In this project, we will use the **State** class and create a binary tree of information. We will use the **TreeNode** and **BinaryTree** classes to store and query **State** objects.

Assignment:

1. Download the **StateTree.zip** file from Mr Greenstein's web site. Unzip it and it will create a **StateTree** directory containing the files **BinaryTree.java**, **State.java**, **states2.txt**, **StateTree.java**, and **TreeNode.java**.

2. Modify the **State.java** file. Create a constructor that sets all of the parameters. Create a **compareTo** method that compares the names of the states. Complete the **getName** and **toString** methods.

3. You are provided a rearranged, out of order set of state data in file **states2.txt**. The **TreeNode.java** file and your **State.java** file should be used as-is.

4. Modify the **BinaryTree.java** provided to handle **TreeNode**s and **State** objects in the list. The **BinaryTree** class must have the following methods.

   loadData() - Reads the states2.txt file and creates the binary tree.
   insert(State next) - Inserts a new **State** object into the binary tree sorted by state name.
   printList() - Prints the tree as a list in ascending order by state name.
   testFind() - Prompts the user for the state name and prints the state's information to the screen.
   size() - Returns the number of nodes in the binary tree.
   clear() - Clears the tree of all nodes.
   printLevel() - Prompts the user for the level of the tree to print, then prints the names of all the states at that level.
   testDepth() - Prints the highest level number (depth) of the binary tree. The root node is level 0. It prints **"Tree empty"** if the tree has no nodes.
   testDelete() - Prompts the user for the state name and deletes the node containing that name.

   Make sure all **String** prompts can handle mixed cases. (e.g. q/Q and Hawaii/HAWAII/hawaii are equivalent)

5. Use the **StateTree.java** user interface, unmodified, to perform operations on the binary tree database.

Hints:
Binary trees lend themselves to neat, short, recursive methods. When implementing any of the **BinaryTree** methods, think about how recursion could be used. For example, the **size** operation could be done by employing two methods, one non-recursive and one recursive. The first method, a non-recursive **public int size()** would be called by the main class **StateTree** and it returns the final tally. A second, recursive **public int size(TreeNode node)** could be called by the first **size()** and it returns the count of the nodes in the subtree below including the **node**. The first **size** only serves to start the recursion and return the final count, while the second recursive **size** does all the counting.

Try the dual non-recursive/recursive methods on **insert**, **printList**, etc. I suggest implementing **testDelete** last.

Here is a sample run output:

```
% java StateTree
Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 2


The tree printed inorder


Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 1

Loading file states2.txt

Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 2


The tree printed inorder
```

```
Alabama              AL     4858979    50744      7    Montgomery           12 14 1819
Alaska               AK     738432     571951     1    Juneau               1  3  1959
Arizona              AZ     6828065    113635     9    Phoenix              2  14 1912
Arkansas             AR     2978204    52068      4    Little_Rock          6  15 1836
California           CA     39144818   155959     53   Sacramento           9  9  1850
Colorado             CO     5456574    103718     7    Denver               8  1  1876
Connecticut          CT     3590886    4845       5    Hartford             1  9  1788
Delaware             DE     945934     1954       1    Dover                12 7  1787
Florida              FL     20271272   53927      27   Tallahassee          3  3  1845
Georgia              GA     10214860   57906      14   Atlanta              1  2  1788
Hawaii               HI     1431603    6423       2    Honolulu             8  21 1959
Idaho                ID     1654930    82747      2    Boise                7  3  1890
Illinois             IL     12859995   55584      18   Springfield          12 3  1818
Indiana              IN     6619680    35867      9    Indianapolis         12 11 1816
Iowa                 IA     3123899    55869      4    Des_Moines           12 28 1846
Kansas               KS     2911641    81815      4    Topeka               1  29 1861
Kentucky             KY     4425092    39728      6    Frankfort            6  1  1792
Louisiana            LA     4670724    43562      6    Baton_Rouge          4  30 1812
Maine                ME     1329328    30862      2    Augusta              3  15 1820
Maryland             MD     6006401    9774       8    Annapolis            4  28 1788
Massachusetts        MA     6794422    7840       9    Boston               2  6  1788
Michigan             MI     9922576    56804      14   Lansing              1  26 1837
Minnesota            MN     5489594    79610      8    St._Paul             5  11 1858
Mississippi          MS     2992333    46907      4    Jackson              12 10 1817
Missouri             MO     6083672    68886      8    Jefferson_City       8  10 1821
Montana              MT     1032949    145552     1    Helena               11 8  1889
Nebraska             NE     1896190    76872      3    Lincoln              3  1  1867
Nevada               NV     2890845    109826     4    Carson_City          10 31 1864
New_Hampshire        NH     1330608    8968       2    Concord              6  21 1788
New_Jersey           NJ     8958013    7417       12   Trenton              12 18 1787
New_Mexico           NM     2085109    121356     3    Santa_Fe             1  6  1912
New_York             NY     19795791   47214      27   Albany               7  26 1788
North_Carolina       NC     10042802   48711      13   Raleigh              11 21 1789
North_Dakota         ND     756927     68976      1    Bismarck             11 2  1889
Ohio                 OH     11613423   40948      16   Columbus             3  1  1803
Oklahoma             OK     3911338    68667      5    Oklahoma_City        11 16 1907
Oregon               OR     4028977    95997      5    Salem                2  14 1859
Pennsylvania         PA     12802503   44817      18   Harrisburg           12 12 1787
Rhode_Island         RI     1056298    1045       2    Providence           5  29 1790
South_Carolina       SC     4896146    30109      7    Columbia             5  23 1788
South_Dakota         SD     858469     75885      1    Pierre               11 2  1889
Tennessee            TN     6600299    41217      9    Nashville            6  1  1796
Texas                TX     27469114   261797     36   Austin               12 29 1845
Utah                 UT     2995919    82144      4    Salt_Lake_City       1  4  1896
Vermont              VT     626042     9250       1    Montpelier           3  4  1791
Virginia             VA     8382993    39594      11   Richmond             6  25 1788
Washington           WA     7170351    66544      10   Olympia              11 11 1889
West_Virginia        WV     1844128    24078      3    Charleston           6  20 1863
Wisconsin            WI     5771337    54310      8    Madison              5  29 1848
Wyoming              WY     586107     97100      1    Cheyenne             7  10 1890

Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
```

```
(8) Print depth of tree
(Q) Quit

Choice ---> 5

Number of nodes = 50

Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 8


Depth of the tree = 10


Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 7


Testing printLevel algorithm

Enter level value to print (-1 to quit)--> 0

Level   0
Delaware

Enter level value to print (-1 to quit)--> 1

Level   1
Connecticut Pennsylvania

Enter level value to print (-1 to quit)--> 2

Level   2
Alabama     New_Jersey  South_Carolina

Enter level value to print (-1 to quit)--> 3

Level   3
```

```
Arkansas    Georgia     New_York    Rhode_Island      Virginia

Enter level value to print (-1 to quit)--> 4

Level   4
Arizona     California  Florida     Massachusetts     New_Mexico  North_Carolina
Vermont     Wisconsin

Enter level value to print (-1 to quit)--> 5

Level   5
Alaska      Colorado    Maryland    New_Hampshire     Ohio  Tennessee   West_Virginia
Wyoming

Enter level value to print (-1 to quit)--> 6

Level   6
Kentucky    Mississippi North_Dakota      Oregon      South_Dakota      Texas Washington

Enter level value to print (-1 to quit)--> 7

Level   7
Indiana     Louisiana   Michigan    Missouri    Oklahoma    Utah

Enter level value to print (-1 to quit)--> 8

Level   8
Illinois    Iowa  Maine Minnesota   Nevada

Enter level value to print (-1 to quit)--> 9

Level   9
Idaho Kansas      Nebraska

Enter level value to print (-1 to quit)--> 10

Level  10
Hawaii      Montana

Enter level value to print (-1 to quit)--> 11

Level  11


Enter level value to print (-1 to quit)--> -1


Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 3
```

```
Testing search algorithm

Enter state name to search for (Q to quit) --> hawaii

Hawaii               HI     1431603    6423       2    Honolulu             8  21 1959

Enter state name to search for (Q to quit) --> louisiana

Louisiana            LA     4670724    43562      6    Baton_Rouge          4  30 1812

Enter state name to search for (Q to quit) --> podunk
Name = podunk  No such state name

Enter state name to search for (Q to quit) --> q
Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 4


Testing delete algorithm

Enter state name to delete for (Q to quit) --> louisiana

Deleted louisiana

Enter state name to delete for (Q to quit) --> q
Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 5

Number of nodes = 49

Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
```

```
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> 2


The tree printed inorder

Alabama              AL     4858979    50744      7    Montgomery           12 14 1819
Alaska               AK     738432     571951     1    Juneau               1  3  1959
Arizona              AZ     6828065    113635     9    Phoenix              2  14 1912
Arkansas             AR     2978204    52068      4    Little_Rock          6  15 1836
California           CA     39144818   155959     53   Sacramento           9  9  1850
Colorado             CO     5456574    103718     7    Denver               8  1  1876
Connecticut          CT     3590886    4845       5    Hartford             1  9  1788
Delaware             DE     945934     1954       1    Dover                12 7  1787
Florida              FL     20271272   53927      27   Tallahassee          3  3  1845
Georgia              GA     10214860   57906      14   Atlanta              1  2  1788
Hawaii               HI     1431603    6423       2    Honolulu             8  21 1959
Idaho                ID     1654930    82747      2    Boise                7  3  1890
Illinois             IL     12859995   55584      18   Springfield          12 3  1818
Indiana              IN     6619680    35867      9    Indianapolis         12 11 1816
Iowa                 IA     3123899    55869      4    Des_Moines           12 28 1846
Kansas               KS     2911641    81815      4    Topeka               1  29 1861
Kentucky             KY     4425092    39728      6    Frankfort            6  1  1792
Maine                ME     1329328    30862      2    Augusta              3  15 1820
Maryland             MD     6006401    9774       8    Annapolis            4  28 1788
Massachusetts        MA     6794422    7840       9    Boston               2  6  1788
Michigan             MI     9922576    56804      14   Lansing              1  26 1837
Minnesota            MN     5489594    79610      8    St._Paul             5  11 1858
Mississippi          MS     2992333    46907      4    Jackson              12 10 1817
Missouri             MO     6083672    68886      8    Jefferson_City       8  10 1821
Montana              MT     1032949    145552     1    Helena               11 8  1889
Nebraska             NE     1896190    76872      3    Lincoln              3  1  1867
Nevada               NV     2890845    109826     4    Carson_City          10 31 1864
New_Hampshire        NH     1330608    8968       2    Concord              6  21 1788
New_Jersey           NJ     8958013    7417       12   Trenton              12 18 1787
New_Mexico           NM     2085109    121356     3    Santa_Fe             1  6  1912
New_York             NY     19795791   47214      27   Albany               7  26 1788
North_Carolina       NC     10042802   48711      13   Raleigh              11 21 1789
North_Dakota         ND     756927     68976      1    Bismarck             11 2  1889
Ohio                 OH     11613423   40948      16   Columbus             3  1  1803
Oklahoma             OK     3911338    68667      5    Oklahoma_City        11 16 1907
Oregon               OR     4028977    95997      5    Salem                2  14 1859
Pennsylvania         PA     12802503   44817      18   Harrisburg           12 12 1787
Rhode_Island         RI     1056298    1045       2    Providence           5  29 1790
South_Carolina       SC     4896146    30109      7    Columbia             5  23 1788
South_Dakota         SD     858469     75885      1    Pierre               11 2  1889
Tennessee            TN     6600299    41217      9    Nashville            6  1  1796
Texas                TX     27469114   261797     36   Austin               12 29 1845
Utah                 UT     2995919    82144      4    Salt_Lake_City       1  4  1896
Vermont              VT     626042     9250       1    Montpelier           3  4  1791
Virginia             VA     8382993    39594      11   Richmond             6  25 1788
Washington           WA     7170351    66544      10   Olympia              11 11 1889
West_Virginia        WV     1844128    24078      3    Charleston           6  20 1863
Wisconsin            WI     5771337    54310      8    Madison              5  29 1848
Wyoming              WY     586107     97100      1    Cheyenne             7  10 1890
```

```
Binary Tree algorithm menu

(1) Read Data from a file
(2) Print the list
(3) Search the list
(4) Delete node
(5) Count nodes
(6) Clear the list
(7) Print the level
(8) Print depth of tree
(Q) Quit

Choice ---> q
```