CS2134 Homework Assignment 6
Spring 2015
Due 3:59 p.m., Monday April. 12, 2015

Your sixth assignment includes both a programming portion and a written portion. The programming portion should consist of a separate file hw06.cpp. The written portion should consist of a separate file, hw06written, saved in a standard document format (.txt, .doc, .htm., or .pdf). Be sure to include your name at the beginning of each file! *You must hand in both files via NYU Classes.*

Programming Part:
A) In this part, you will store a list of correctly spelled words and a *point value* associated with each word into a variable of type `map<string, int>`. You will perform the following steps:

- Enter the point value from a file called `Letter_point_value.txt` for each letter into a variable of type `vector<int>`. The point value associated with 'A' goes into position 0, and 'B' goes into position 1, etc.
- The point value of a word is determined by the point value of each letter. To compute the point value of a word, add up the point values of each letter in the word. For example, the point value of "cat" is 4 + 1 + 1 = 6, since 'C' has a point value of 4, 'A' has a point value of 1, and 'T' has a point value of 1. Upper and lower case letters have the same point value. (e.g. So "CAT" also has a point value of 6.)

  Create a function to compute the point value of a word.
- The words you will store are in a file called `ENABLE.txt`. You will read in each word and store it and its associated point value in a variable of type `map<string,int>`.

B)[1] Use the recursive function from the extra credit problem[2] written in homework assignment #4, where the user enters a string and you find all the combinations of the string.

For each string created from the recursive function, you test to see if it is in the `ENABLE` word list. If so, you print out the word and the points associated with the word.[3] Use the `map<string,int>` you created in programming part **A**.

C) Write a generic function template called `Advance` that takes two arguments. The first argument is an iterator which has the capabilities of a forward iterator. The second argument is an `int`. The function does not return any value, but its iterator argument now points `n` items ahead.

The signature is:

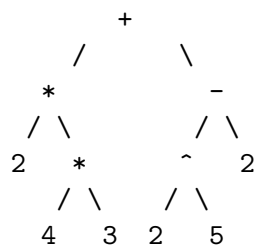
```
template <class ForwardIterator>
  void advance (ForwardIterator& it, int n);
```

Add preconditions and postconditions as comments above your generic function template.

[1] Please look on Piazza for hints on how to solve this problem.
[2] If you adapt the published solution, please cite you are doing so.
[3] This program could help you if you play `Words With Friends`.

Written Part:

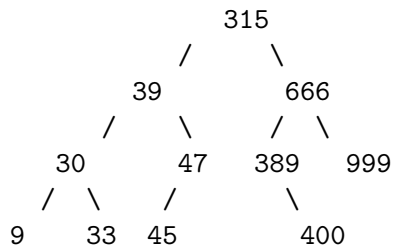
1. Run times:
   (a) what is the running time of programming question **A**?
   (b) if in programming question **A** you had used an `unordered_map` instead of a `map` what would be the running time?[4]
2. What does the height of a binary search tree mean in relation to its searching efficiency?
3. How many different binary trees can be made from three nodes that contain the values 1, 2, 3?
4. For the following expression tree:



   (a) what is the output of an inorder traversal of the tree
   (b) what is the output of a preorder traversal of the tree
   (c) what is the output of a postorder traversal of the tree
5. For the tree in written problem 4 determine:
   (a) which node is the root
   (b) which nodes are the children of the root node
   (c) which node is the sibling of the node containing 4
   (d) which nodes are leaves
   (e) the tree's depth
   (f) the tree's size
   (g) the size of node `'-'`
   (h) the height of node `'-'`
   (i) the depth of node `'-'`
6. Given the implementation of the binary search tree presented in class, what is the best order to insert the following numbers $\{0, 1, 2, 3, 4, 5, 6, 7\}$ so that the tree has:
   - minimal height. Show the tree that would be created if they were added in that order.
   - maximal height. Show the tree that would be created if they were added in that order.
7. (a) Show the result of inserting $(2, 1, 4, 5, 8, 3, 6, 7)$ into an initially empty binary search tree.
   (b) What is the output of an inorder traversal of your tree obtained in (a).
   (c) What is the output of a preorder traversal of your tree obtained in (a).
   (d) What is the output of a postorder traversal of your tree obtained in (a).

[4]The `map` class is often implement using a balanced binary search tree.

8. Consider the following binary search tree:



(a) Show what happens when 315 is removed.

(b) Show what happens when 39 is removed (to original tree, not the tree after 315 was removed).