Subject: SPECIFICATIONS
Posted by btlindow on Sat, 17 Sep 2016 03:51:20 GMT

I thought I would make a thread specifically for project specifications and questions pertaining to specifications. That way they are all relatively easy to find in one thread.

:d

Subject: Re: SPECIFICATIONS
Posted by lusth on Sat, 17 Sep 2016 16:39:43 GMT

I gave a similar project last semester featuring red-black trees.

Things that were missed then:

* forgetting to put a level number in front of the level
* forgetting to put an X designator for the root
* not using strcmp to order the keys
* forgetting to put the balance indicator wherever a node name appears

I will be using an AVL checker to check your trees, so the syntax has to match exactly.

I'm going to post an unfinished version of the checker soon. You should finish it out and check some hand drawn trees BEFORE you start your AVL code.

Subject: Re: SPECIFICATIONS
Posted by btlindow on Sun, 18 Sep 2016 06:31:14 GMT

Is it acceptable to use an in-order traversal via recursion to determine the level of the nodes in our trees?

Subject: Re: SPECIFICATIONS
Posted by lusth on Sun, 18 Sep 2016 23:17:37 GMT

Only if your entire display of the tree is O(n).
Just to make sure that I am understanding this right and that I don't go too far before I have to turn around...

My traversal takes O(n) time and displaying the level order of my tree takes O(n) time. Together that is a combined time of O(n) + O(n). Which, if I recall correctly, is still O(n).

Is my thinking correct? I could theoretically do a traversal a dozen times and still have it be O(n)?

I am using one traversal to determine the levels of each node, the max weight of each node (furthest distance to a leaf), and min weight of each node (shortest path to a leaf).

Seems n log n to me. log n levels and n time to find all nodes at that level.

Even if I am doing a constant amount of work at each level?

My traversal only visits each node once and does a constant amount of work at that node. Intuitively, to me, visiting n nodes and doing constant work would be O(n) time.

\[ T(n) = 2T(n/2) + 1 \]
\[ n^{\log base 2 of 2} = n^{1} = n \]
\[ n > 1 \]

Case 1

Theta (n).

If I were to linear work at the level I could see it being n log n. Maybe I am doing this wrong?

In the project specs you have "Girls' night out!" which turns into girls night out. Just to be clear, you insert each of these words (girls,night,out) into the tree, not the phrase "girls night out" correct?
Double quoted phrases are inserted in their entirety, so "girls night out" would be a single key.

In the example of legal input from a file, after 'i' there is a string with double quotes to be read. After 'f', which calls the function to report the frequency of a word or phrase, there are no double quotes. Will there be no double quotes if there is only one word to be read? And double quotes if it is a phrase?

The double quotes are optional for a single token. Since the scanner's readString does not return the double quotes in the string that it reads, your program doesn't need to know if there are spaces in the string or not. I have a cleaning function that removes redundant whitespace. Without having to do anything special, that cleaning function "does the right thing" on strings with no whitespace. So everything works if I have:

i "Squarepants"

or if I have:

i Squarepants

They are the same token.

When generating the statistics for the command r, should we consider a node with, lets say, a frequency of 100 as 1 node or 100 nodes?
That would be one node.

Will the command line arguments be in the order shown in the example in the instructions or is it possible they could be in a different order?

When outputting the report or the frequency of a word is there a given format you want us to use?

berryhilleric wrote on Thu, 06 October 2016 16:42
Will the command line arguments be in the order shown in the example in the instructions or is it possible they could be in a different order?

In the order shown.

bpsokol wrote on Thu, 06 October 2016 16:49
When outputting the report or the frequency of a word is there a given format you want us to use?

I'm not sure what you are asking. The spec indicates the output for each command.

The spec says:
"the number of nodes in the tree"
the distance from the root to the closest null child pointer (the root is at distance 0)
the distance from the root to the furthest null child pointer (the root is at distance 0)

So when running the report should we output "the number of nodes in the tree is n" or output just n?
Also when reporting the frequency the spec says "f W report the frequency of word or phrase W"
so what would our output look like for this?

---

Subject: Re: SPECIFICATIONS
Posted by lusth on Fri, 07 Oct 2016 12:19:45 GMT

Any reasonable output will be accepted. Printing just 3 numbers for the statistics would not be reasonable, since there is no explanation of what these three numbers mean.

---

Subject: Re: SPECIFICATIONS
Posted by mlpearson4 on Sat, 08 Oct 2016 00:32:12 GMT

Just to double check, "   " should be inserted as "    " correct? In other words, if you’re only inserting quoted string of all spaces with no letters, a single space would be inserted.

EDIT: The forum removes the 5 spaces in the first space string.

---

Subject: Re: SPECIFICATIONS
Posted by lusth on Sat, 08 Oct 2016 13:13:59 GMT

Correct, double quoting preserves whitespace, with contiguous white space characters collapsed to a single space. For example
"hello
    world"becomes

"hello world"Note that the opening and closing quotes are not part of the string that is inserted.

---

Subject: Re: SPECIFICATIONS
Posted by AdamJAled on Thu, 13 Oct 2016 00:31:55 GMT

When we do swaps before we call our deleteFixUp function for our AVL trees, do you have a preference on whether we should do Predecessor or Successor first, or do you not care?
That's up to you, since either is legit.

Just to be completely clear, there is never a situation where we should insert an empty string into a tree? -- and by empty I mean a null string or a string consisting of just a space or spaces.

If it's a quoted string, then a string containing a single space could end up in the tree.

In the c style guide it states that the line length should not exceed 80 characters. Is this a strict boundary or can it be slightly longer to improve readability?

A typical terminal window starts out at 80 characters per line. That's the reason for the restriction, so that there is no line wrapping.

What should the run time be for the statistics command? More specifically, what should the run time be for calculating the distance from the root to the closest/furthest node with a null child?
Another question: For the -b option, since we're using a standard bst, we don't need to display each node's favoritism, since that doesn't really apply, correct? I don't see it explicitly saying we don't need to in the project guidelines, but the given example doesn't.

My finding of the distances to a null child takes O(n) time.

BST nodes don't have favorites, so favorite markers should not be printed.

The project specifications say "Note that the parent of the root is itself". Will points be deducted for having NULL as the parent of the root instead?

I've implemented BS and AVL trees as completely separate modules, as opposed to having AVL trees call BST functions. Would this be considered an "unreasonable amounts of duplicated code" and subject to a "serious" point deduction?

Regarding the stylistic 80 character limitation, what do you do if you just need a really long string? Adding newlines for readability would also add them to the string, and using extra code to append to the string just seems overkill. Is it ok to have a string assignment or a printf statement exceed 80 characters?
Einhaender wrote on Fri, 28 October 2016 21:27
The project specifications say "Note that the parent of the root is itself".
Will points be deducted for having NULL as the parent of the root instead?

As long as the display shows the parent of the root is the root, I don't care what you store in the parent pointer of the root.

Einhaender wrote on Fri, 28 October 2016 21:35
I've implemented BS and AVL trees as completely separate modules, as opposed to having AVL trees call BST functions. Would this be considered an "unreasonable amounts of duplicated code" and subject to a "serious" point deduction?

Short answer, no. Long answer, ask me this in class Tuesday.

sestephens wrote on Sat, 29 October 2016 17:09
Regarding the stylistic 80 character limitation, what do you do if you just need a really long string? Adding newlines for readability would also add them to the string, and using extra code to append to the string just seems overkill. Is it ok to have a string assignment or a printf statement exceed 80 characters?

Sometimes, you just can't help a long line of code. In the case of C, though, you can break strings apart easily. These three printfs are equivalent:
\begin{verbatim}
printf("a very long string");
printf("a very " "long string");
printf("a very \\
   "long string");
\end{verbatim}
C automatically concatenates two or more adjacent strings.