CS101 Algorithms and Data Structures

Fall 2021

Homework 6

Due date: 23:59, November 7, 2021

1. Please write your solutions in English.

2. Submit your solutions to gradescope.com.

3. Set your FULL NAME to your Chinese name and your STUDENT ID correctly in Account Settings.

4. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.

5. When submitting, match your solutions to the according problem numbers correctly.

6. No late submission will be accepted.

7. Violations to any of the above may result in zero grade.

1: (12') Multiple Choices

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 1 point if you select a non-empty subset of the correct answers.

Note that you should write you answers of section 1 in the table below.

Question 1	Question 2	Question 3	Question 4	Question 5	Question 6
C	BD	BD	ACD	A	ABCD/AC

Question 1. *Which of the followings are true?*

(A) *For a min-heap, in-order traversal gives the elements in ascending order.*

(B) *For a min-heap, pre-order traversal gives the elements in ascending order.*

(C) *For a BST, in-order traversal gives the elements in ascending order.*

(D) *For a BST, pre-order traversal gives the elements in ascending order.*

Question 2. *For a Binary Search Tree (BST), which of the followings are true?*

(A) *If we erase the root node with two children, then it will be replaced by the maximum object in its right sub-tree.*

(B) *The cost for erasing the root node who has two children is $O(n)$.*

(C) *In a BST with N nodes, it always takes $O(\log N)$ to search for an specific element.*

(D) *For a BST, the newly inserted node will always be a leaf node.*

Question 3. *Suppose we want to use Huffman Coding Algorithm to encode a piece of text made of characters. Which of the following statements are true?*

(A) *Huffman Coding Algorithm will compress the text data with some information loss.*

(B) *The construction of binary Huffman Coding Tree using priority queue has time complexity $O(nlogn)$, where n is the size of the character set of the text.*

(C) *When inserting nodes into the priority queue, the higher the occurrence/frequency, the higher the priority in the queue.*

(D) *The Huffman codes obtained must satisfy prefix-property, that is, no code is a prefix of another code.*

Question 4. *Which of the following statements are true for an AVL-tree?*

(A) *Inserting an item can unbalance non-consecutive nodes on the path from the root to the inserted item before the restructuring.*

(B) *Inserting an item can cause at most one node imbalanced before the restructuring.*

(C) *Removing an item in leaf nodes can cause at most one node imbalanced before the restructuring.*

(D) Only at most one node-restructuring has to be performed after inserting an item.

Question 5. *Consider an AVL tree whose height is h, which of the following are true?*

(A) This tree contains $\Omega(\alpha^h)$ nodes, where $\alpha = \dfrac{1+\sqrt{5}}{2}$.

(B) This tree contains $\Theta(2^h)$ nodes.

(C) This tree contains $O(h)$ nodes in the worst case.

(D) None of the above.

Question 6. *Which of the following is TRUE?*

(A) The cost of searching an AVL tree is $O(\log n)$ but that of a binary search tree is $O(n)$

(B) The cost of searching an AVL tree is $O(\log n)$ but that of a complete binary tree is $O(n \log n)$

(C) The cost of searching a binary search tree with height h is $O(h)$ but that of an AVL tree is $O(\log n)$
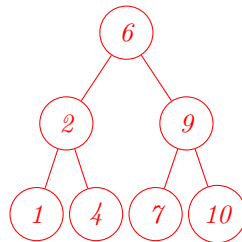
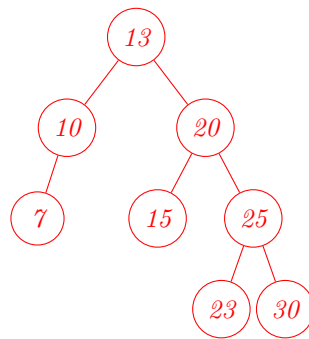*(D) The cost of searching an AVL tree is $O(nlogn)$ but that of a binary search tree is $O(n)$*
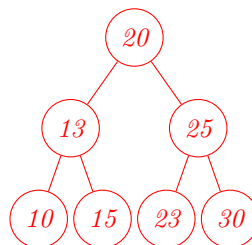
2: (3'+3'+2'+3') BST and AVL Tree

Question 7. *Draw a valid BST of minimum height containing the keys 1, 2, 4, 6, 7, 9, 10.*

Question 8. *(1) Given an empty AVL tree, insert the sequence of integers $15, 20, 23, 10, 13, 7, 30, 25$ from left to right into the AVL tree. Draw the final AVL tree.*

(2) For the final AVL tree in the question (1), delete 7. *Draw the AVL tree after deletion.*

(3) For an AVL tree, define D = the number of left children - the number of right children, for the root. Then what is the maximum of D for an AVL tree with height n ?

Ans: An AVL tree with height n at most has $2^{n+1} - 1$ nodes, assume an AVL tree with height n at least has $f(n)$ nodes, then f(n)=f(n-1)+f(n-2)+1, when n>2, and f(1)=2,f(0)=1

solve f(n): f(n)+1=(f(n-1)+1)+(f(n-2)+1) let g(n)=f(n)+1 ,

then g(n) is Fibonacci numbers, $g(n) = g(n-1) + g(n-2)$ g(0)=2,g(1)=3 , the equation $x^2 - x - 1 = 0$ has two solutions,$\frac{1-\sqrt{5}}{2}, \frac{1+\sqrt{5}}{2}$,then $g(n) = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^{n+3} - (\frac{1-\sqrt{5}}{2})^{n+3}]$,so $f(n) = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^{n+3} - (\frac{1-\sqrt{5}}{2})^{n+3}] - 1$

Then D=the number of left children - the number of right children=$2^{n+1} - 1 - \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^{n+3} - (\frac{1-\sqrt{5}}{2})^{n+3}] + 1$=$2^n - \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1}]$

3: (3'+3') Huffman Coding

After you compress a text file using Huffman Coding Algorithm, you accidentally spilled some ink on it and you found that one word becomes unrecognizable. Now, you need to recover that word given the following information:

**Huffman-Encoded sequence of that word:**

000101001110110100

Frequency table that stores the frequency of some characters:

characters	a	b	e	g	l	o	r
frequency	2	3	3	4	1	4	2

Question 9. *Please construct the binary Huffman coding tree according to the given frequency table and draw the final tree below.*
Note: The initial priority queue is given as below. When inserting nodes into the priority queue, the priority of nodes with the same frequency follows "last in least priority".

l	a	r	b	e	g	o
1	2	2	3	3	4	4
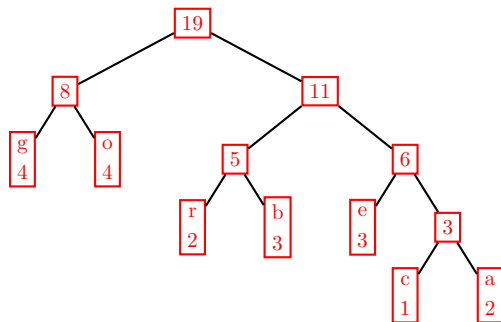
Question 10. *Now you can "decompress" the encoded sequence and recover the original word you lost. Please write the original word below.*
googler

4: (9')Only-child

We define that the node is only-child if its parent node only have one children(Note: The root does not qualify as an only child). And we define a function for any binary tree T : $OC(T) =$ the number of only-child node.

Question 11. *(3')Prove the conclusion that for any nonempty AVL tree T with n nodes, $OC(T) \leq \frac{1}{2}n$.*

Ans: In an AVL tree just the leaves may be only-children, and therefore for every only-child in T, there exists a unique parent node that is not an only-child. The total number of only children in T is at most n/2

Question 12. *(3') For any binary tree T with n nodes, Is it true that if $OC(T) \leq \frac{1}{2}n$ then $height(T) = O(\log n)$? If true, prove it. If not, give a counterexample.*
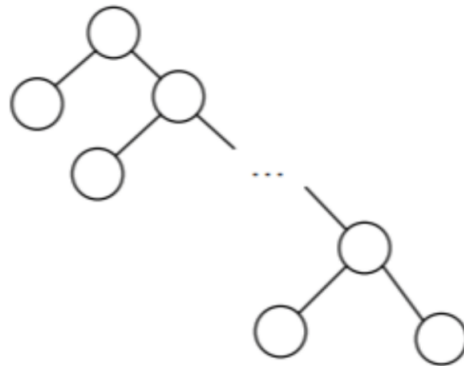
Ans: A counterexample

Question 13. *(3') For any binary tree T, Is it true that if there are n_0 only-children and they are all leaves, then $height(T) = O(\log n_0)$? If true, prove it. If not, give a counterexample.*

Ans: A counterexample