Theory Assignment 4

CS 452/652/752 Advanced Algorithms and Applications

Full points: 100 + Extra points: 20

Question 1 - 10 points

Fractional Knapsack Problem: Use an example to show that the Fractional Knapsack Problem has both greedy-choice and optimal substructure property.

Question 2 - 15 points

Huffman Compression: Suppose you are given a dataset that contains 599,643 characters. The characters can only be lowercase letter from a to z. (1) Given their number of occurrence (frequency) in the dataset, build the Huffman tree and list the Huffman code for each character. (2) Assume storing each character in its original form takes 8 bits, what is the total cost (total number of bits) for storing the 599,643-character dataset with and without Huffman compression?

Character	Frequency	Character	Frequency	Character	Frequency
a	54024	j	724	s	37575
b	8414	k	4782	t	48165
c	13896	l	16726	u	22030
d	28041	m	42380	v	5199
e	74809	n	15298	w	14113
f	12530	o	46499	x	710
g	13559	p	9957	y	12177
h	38961	q	215	z	667
i	41005	r	37187		

Question 3 - 10 points

Amortized Analysis: Suppose we have an implementation of dynamic array. The array's initial size is 0. New elements are always appended to the back of the array. If the array is full, we create a new array with twice of the size of the current array and copy old elements to the new array. The old array is discarded. Use amortized analysis to show that the average cost of performing each insertion operation is $O(1)$.

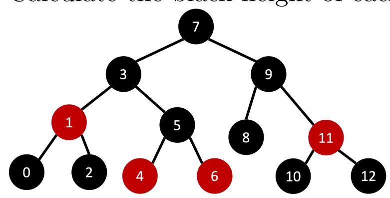
Question 4 - 10 points

Binary Search Tree: Build a binary search tree using the following elements for the two element insertion orders. (1) Inserting beginning from the left most element. (2) Inserting beginning from the right most element.

$$45, 20, 30, 89, 60, 34, 50, 52, 73, 41, 9, 2, 23, 52, 68, 36$$

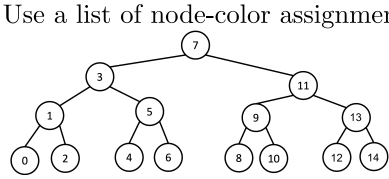
Question 5 - 10 points

Calculate the black-height of each node in the following red-black tree.



Question 6 - 10 points

Color the following tree to make it a red-black tree. Show three ways of coloring such that the black-height of the resulting red-black trees are 2, 3, and 4. Use a list of node-color assignment to show each way of coloring.



Question 7 - 10 points

What are the running time of search, insert, delete, minimum, maximum, successor, and predecessor operations of red-black tree?

Question 8 - 10 points

What are the running time of search, insert, minimum, successor operations of proto-vEB tree and min/max-vEB tree ?

Question 9 - 10 points

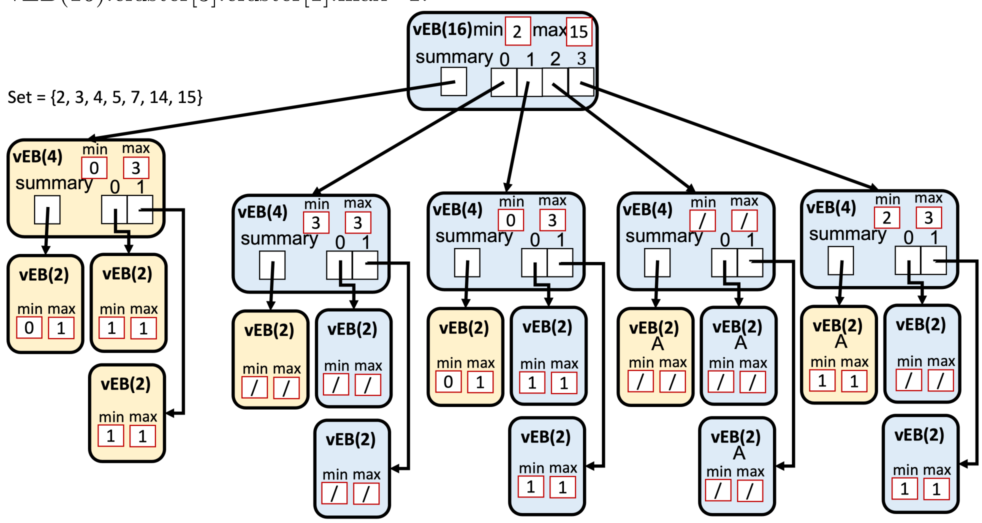
Solve the following recurrence $T(n) = 4T(\sqrt{n}) + O(lg\sqrt{n})$

Question 10 - 10 points

(1) Which parts of the min/max-vEB tree need to be updated when inserting key $x = 10$?

(2) After $x = 10$ is inserted, which parts of the min/max-vEB tree need to be updated when inserting key $y = 13$. Use notations in the Cormen book to identify each part.

For example, you should specify updates as a series of value assignments to components in the tree: vEB(16).min = 2, vEB(16).summary.cluster[0].min=1, and vEB(16).cluster[3].cluster[1].max=1.



Question 11 - 15 points

Using the terminologies in the Cormen book, how are the following operations implemented for a given root of a min/max-vEB tree V

1. Check if the subtree V is empty or not.
2. Find the entry in V's summary that corresponding to the cluster that key x belongs. Mark that entry "1".
3. Among the clusters of V, find the cluster that contains the successor of a given key x. If not found, return NIL.