CS203 DSAA Fall 2021 Quiz 2

南方科技大学

Problem 1 [20 points] Time Complexity of Heap Building

The time complexity of turn sized-n array A into a binary heap on S via root-fix operator on dynamic array is O(n), where A stores the values in set S.

18

Without loss of generality, suppose the heap is a full tree.

List the number of nodes and necessary time consumption for each level:

Level	# of nodes	time consumption
0	2^0	h
1	2^1	$h-1$
2	2^2	$h-2$
⋮	⋮	⋮
$h-2$	2^{h-2}	2
$h-1$	2^{h-1}	1
h	2^h	0

So the overall time consumption is:

$$F = 2^0 h + 2^1(h-1) + 2^2(h-2) + \cdots + 2^{h-2}(2) + 2^{h-1}(1)$$

then $$2F = 2^1(h) + 2^2(h-1) + \cdots + 2^{h-2}(3) + 2^{h-1}(2) + 2^h(1)$$

Substract these two equation:

$$F = -h + 2^1 + 2^2 + \cdots + 2^{h-1} + 2^h = 2^{h+1} - 2 - h$$

Since it's a full tree, then number of all nodes is $\sum_{i=0}^{h} 2^i = 2^{h+1} - 1 = n$

Hence $$F = n - 1 - h = O(n)$$

$\Rightarrow$ The time complexity is $O(n)$.

Problem 2 [20 points] Height of Balanced Binary Search Tree

A balanced binary search tree with n nodes has height O(log n).

+17

Just need to find the greatest height of a AVL Tree.

According to the property of BBST, suppose height of h has at most $T(h)$ nodes, then

$T(h) = T(h-1) + T(h-2) + 1$, $T(0) = 1$, $T(2) = 2$, $T(3) = 4$

~~so~~ restate as: $a_n = a_{n-1} + a_{n-2} + 1$, so $a_{n+1} = a_n + a_{n-1} + 1$

~~Substrat~~ Substract these two equations: $\underbrace{a_{n+1} - a_n}_{b_n} = \underbrace{(a_n - a_{n-1})}_{b_{n-1}} + \underbrace{(a_{n-1} - a_{n-2})}_{b_{n-2}}$

With initial conditions $b_0 = 1$, $b_1 = 2$.

Plug in Fibonacci series, we get

$$b_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right]$$

Add b_0 to b_{n-1}, we have:

$$a_n = b_{n-1} + b_{n-2} + \cdots + b_1 + b_0 + a_0$$

so $$a_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n} + \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + \cdots + \left(\frac{1+\sqrt{5}}{2}\right)^{1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1} - \cdots - \left(\frac{1-\sqrt{5}}{2}\right)^{1}\right] + 1$$

$$a_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+3} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+3}\right] - 1$$

So $$T(h) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{h+3} - \left(\frac{1-\sqrt{5}}{2}\right)^{h+3}\right] - 1$$

solve $T(h) = n$:

$$\left(\frac{1+\sqrt{5}}{2}\right)^{h+3} - \left(\frac{1-\sqrt{5}}{2}\right)^{h+3} = \sqrt{5}n + 1 \geq \left(\frac{1+\sqrt{5}}{2} - \frac{1-\sqrt{5}}{2}\right)^{h+3} = \sqrt{5}^{h+3}$$

so $h \leq \log_{\sqrt{5}}(\sqrt{5}n + 1) - 3 = O(\log n)$

Problem 3 [30 points] Huffman Encoding

Given (character, frequency) pairs as following:

H	N	S	O	E	Y	T	D
14	16	8	12	30	18	43	65

(a) Show the detail steps of building its Huffman tree, i.e., draw the Huffman tree building process step by step

(b) Write down the corresponding scheme of the Huffman tree you obtained in (a), you only need draw a table, which contains two columns, the left is the character, the right is its corresponding Huffman coding

(c) Write down the corresponding codes of string "HONESTY".

+30

(a). S O H N Y E T D

8 12 14 16 18 30 43 65

new nodes: 20 30 38 60 81 125 206

① pick S and O, get 20

② pick H and N, get 30

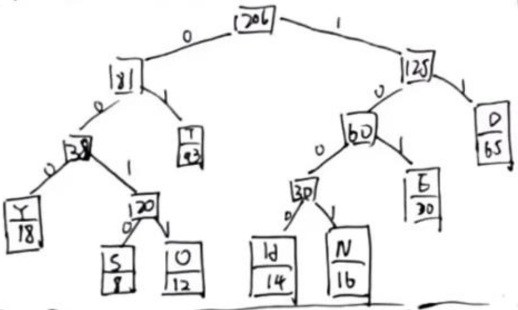
③ pick Y and 20, get 38

④ pick 30 and E, get 60

⑤ pick 38 and T, get 81

⑥ pick 60 and D, get 125

⑦ pick 81 and 125, get 206.

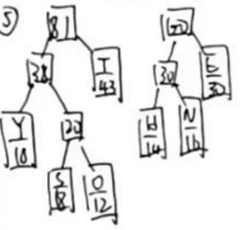


(b).

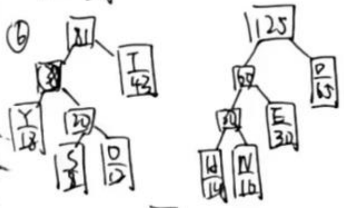
S	0010
O	0011
H	1000
N	1001
Y	000
E	101
T	01
D	11

(c) 1000 0011|1001|101|0010|01|000

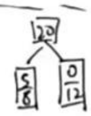
⑤



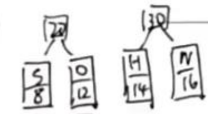
⑥



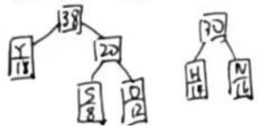
①



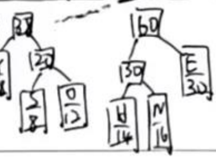
②



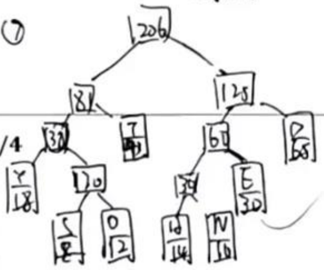
③



④



⑦



Problem 4 [30 points] AVL-Tree

Let us define a binary search tree as following:

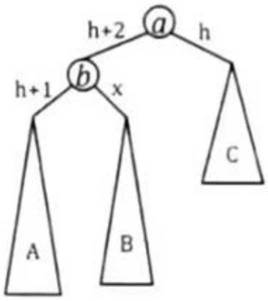


Figure 1. left-left case

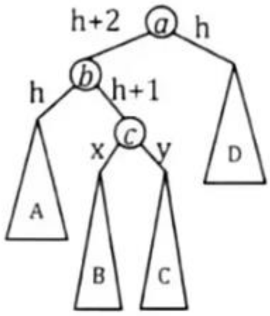


Figure 2. left-right case

(a) Given the imbalance node a in Fig. 1, after remedied the imbalance node a,

a→leftchild = B, a→rightchild = C,

b→leftchild = A, b→rightchild = a,

(b) Given the imbalanced node a in Fig. 2, after remedied the imbalanced node a,

a→leftchild = C, a→rightchild = D,

b→leftchild = A, b→rightchild = B,

c→leftchild = b, c→rightchild = a.

(c) Draw the corresponding balanced binary search tree of Figures 1 and 2.

Fig 1:

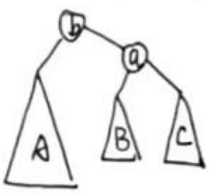
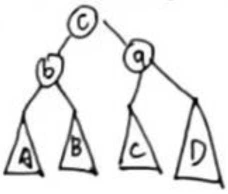


Fig 2:



(d) Given the following imbalance BBST, please draw the balanced BBST after remedy