Answer Pages

Question 21 (pushAll) answer:

tord pushan

void Tree: pushAll (TreeNode * n) {

if (n == NULL) ? "don't need modify tree, just push data return i?

// push and call recursion st. push (temp); push All (n > left Child); ?

Question 22 (KStep) answer:

Tree: KStep() 3

pushAll (root);

while (St. & Empty () != true) {

pushAll (st. pop () > right); }



Question 23 (hasMore) answer:

treeNode * temp = root;

f(root!=NUL) \quad \temp = root;

While(temp > right!= NUL) \quad \temp = temp > right; \(\temp \) \data == \st \rightarrow \data \quad \quad \temp \\

return true; \quad \quad \temp \quad \temp \quad \



Question 24 (step1) answer:

mt Tree: step1();

treeNode * temp= st. pop();

int out = temp> data;

veturn out; ?

Question 25 (step1 running time) answer:

t Tree: Step (Int K) 3

treeNode * temp = st. top();

for (Int i=0, i < k, i+1) 3

If (st. Elmpty ()!=true) 3

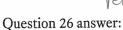
temp = st. pop(); 3

return; 39

If (st. is Empty ()!=true) 3

Fut out= st. top > data;

return out; 4



Lower Bound	0(1)
Average	Oliogn)
Upper Bound Case	Oln



```
Question 27 (buildPerfectTree) answer:
Quad Tree Node *
                     QuadTree = : buidPerfectTree (int k 126BAPIXEL P) 3
             QuadtreeNode * temp = new anadtree Node();

Temp build Helper (0,0, pow(2, K), p, root);
The Quad Tree: (Int itum, it, power) &
          int but = 1;
          for (int 1=0; ) <= k; 1++19
               but=out* num; 3
          return out;
QuadTroeNode & QuadTroesbuildHelper(intx, inty, int size, KGBAPixel p, QuadTreeNode curr) &
            dear (root),
            if (resolution == 1)3
                curr = new Quad tree Node ();
                curr > element = P; 9
            build Helper (x, y, size/z, p, curr> nw Child);
            buildHelper(x+size/2, y, size/2,p, curr>reChild);
            buildHelper (8, y+size/2, p, um> suchild);
            buildHelper(otaize/2, ytsize/2, p, curr=sechild);
                Question 28 (perfectify) answer:
          Quad tree :: perfectify ( int levels ) &
 Word
               mt Longht = getHamph (root);
               perfectify ( level - height, root);
         Quadtree: getheight (Quadtree Node + curr) &
  M
               INT T=0
               if ( owr == NULL) 3
                     return 119
               V= 1+ gotheralt (corr + wwchild); 9
         Quadtree = perfectiony ( Int num, Quadtree Node * & ourr) ?
 bion
                 if ( curr -> nuchild == NUUL) 3
                 curr > nwchild = buildhelper (num, p))
                  cur -> ne child = build helper (num ip) )
                  aur -> suchild = build helper (www 1P);
                  cur- se child = build letper (num, p); 9
                 perfectiaty (num, our-snuchild);
                  perfectional ( num, aur > nethild);
               Question 29 (perfectify running time) answer:
                 perfect ticity ( hum, cur > SNChild);
                 perfecticity (mm, cur > sechild);
            Exam copy 307
                                           Page 14
```

a)

b)

c)

d)

e)

f)

Question 30 answer:

You may answer this question by filling in these blanks, or use the blank space for your own proof/disproof.



Preliminaries Let H(n) denote the maximum height of an n-node SAVL tree, and let N(h) denote the minimum number of nodes in an SAVL tree of height h. To prove (or disprove!) that $H(n) = \mathcal{O}(\log n)$, we attempt to argue that

$$H(n) \leq 3\log_2 n$$
, for all n

Rather than prove this directly, we'll show equivalently that

$$N(h) \ge \underline{\qquad}$$
, (1pt)

Proof For an arbitrary value of h, the following recurrence holds for all SAVL Trees:

We can simplify this expression to the following inequality, which is a function of N(h-3):

$$N(h) \ge \underline{\hspace{1cm}} \times \underline{\hspace{1cm}} \times \underline{\hspace{1cm}} (1pt)$$

By an inductive hypothesis, which states:

$$N(h) \ge 2^{h/3}$$
, $N(0) \ge 2^0$, $N(0) \ge 2^{1/3}$, $N(2) \ge 2^4$, (1pt)

we now have

$$N(h) \ge \frac{\sqrt{3}}{2}$$
 = part (a) answer, (1pt)

which is what we wanted to show.

Given that $2^0 = 1, 2^{1/3} \approx 1.25$, and $2^{2/3} \approx 1.58$, what is your conclusion?

Is an SAVL tree $\mathcal{O}(\log n)$ or not? (Circle one): (2pt)

Overflow Page

Use this space if you need more room for your answers.

