CSL 201 Data-Structures Minor Test 2 7th Oct, 2013

Name ____________________ Entry Number ____________________ Gp No. ______

Your Lab day ____________________ Your TA ____________________

Answer all the questions in the space provided for each question. You can use the last page for rough work.

1. [5] What is the maximum and the minimum number of nodes in an AVL tree with 7 levels ?

Maximum number of nodes in AVL tree with 7 levels $= 2^7 - 1 = 127$ nodes

Minimum no of nodes: level 1 → 1

The tree will be (one option) like this

2 → 2

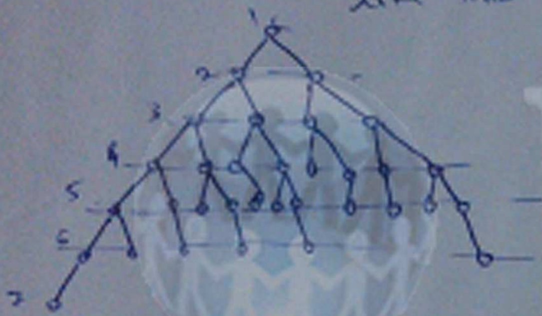
3 → $2^2 = 4$

4 → $2^3 = 8$

5 → 11

6 → 6

7 → 1



Minimum no = 33

5

2. [5] A company maintains records of its customers in an array. The record of each customer contains the name, time stamp indicating the date when he became a customer, and the order placed by the customer. The entries in the array are stored in alphabetical order of the names of the customers. Currently there are around 80,000 customers. The company wants to give a special gift to its 4th customer in order of placing orders with the company. Indicate the most efficient way of picking up the winner from the array.

Assuming that order of placing orders is same as order of time stamp.

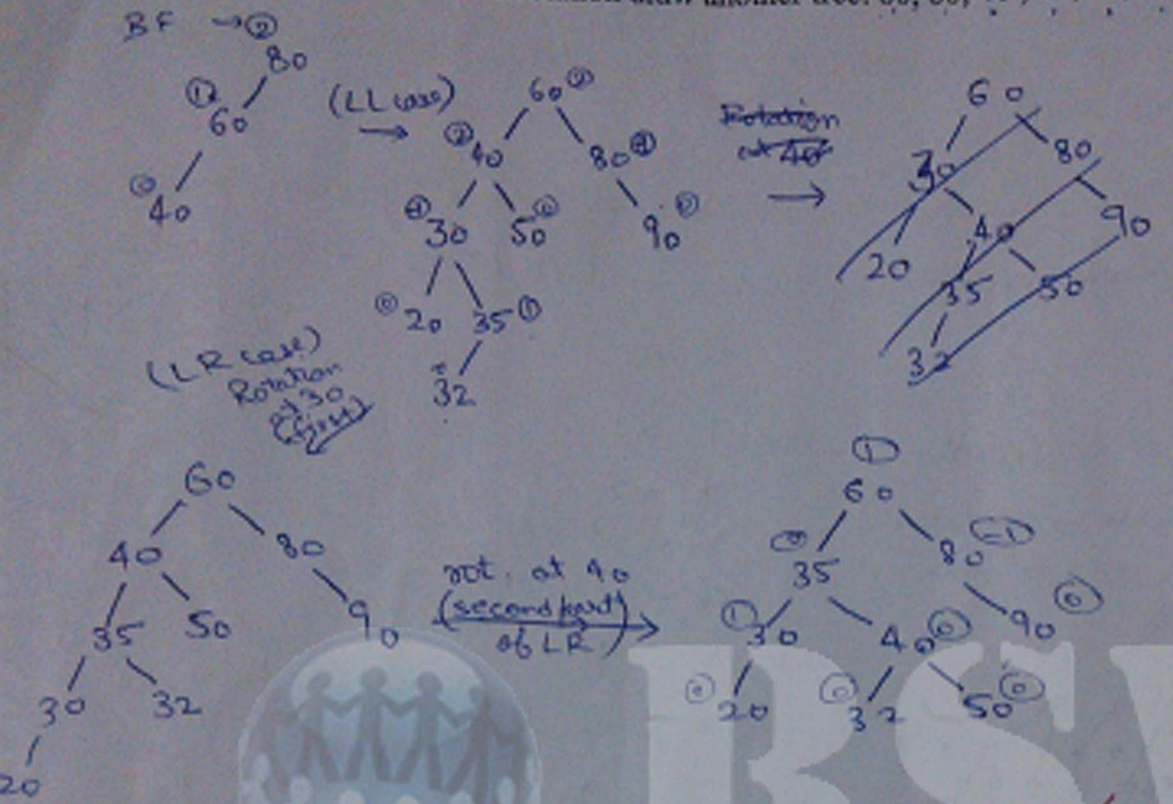
We pick first four customers, create ~~max~~ heap acc to time st

if next element is less that max element, we replace & heapify else go on to next element. At last the ~~head~~ is change to min heap & take ~~last~~ first ele

5

3. **[6]** Draw an AVL tree by inserting the following elements in the given order. Whenever the structure of the tree needs a rotation draw another tree: 80, 60, 40 , 30, 90, 50, 20, 35, 32.

6



The number in circle is Balance factor = height(left) − height(right)

Balanced

4. **[4]** We want to put 30,000,000 records in a B-tree of order 230. What is the maximum height of the B-tree ?

The ~~maximum~~ height of a B-tree ~~with~~ of order 230 with 30,000,000 records is

$$h = \lceil \log_{230}(30{,}000{,}000) \rceil = \lceil 3.165 \rceil = 4$$

$\lceil \frac{m}{2} \rceil$ 0.5

5. **[6]** Sort the following array using the quick algorithm that we did in the class. You should use the first element as the pivot. Show the arrays that are formed during the algorithm:
[45 3 76 32 12 92 34 9 56 60 20]

Keeping 45 as pivot

[45 3 76 32 12 92 34 9 56 60 20]

i→ i 20 →j 9 j 92 76

(as 45 will be swapped with 34, the following array is formed) →i

[34 3 20 32 12 9] [45] [92 56 60 20]

~~[9] 3 20 32 12 9] [34] [45] [20 56 60 [92]]~~

~~[[3] [9 12] [20] [32 34]~~

3

[[9 3 20 32 12] [34] [45] [76 56 60] [92]]

[[[3] [9] [20 32 12]] [34] [45] [60 [56] [76]] [92]]

[[[3] [9] [[12] [20] [32]] [34] [45] [56] [[60] [76]] [92]]

⇒ [[3 9 12 20 32 34] [45] [56 60 76 92]]

⇒ [3 9 12 20 32 34 45 56 60 76 92]

6. **[6]** Solve the following recurrence relation. Continue to next page if necessary.
T(n)=2 T(n/5) + n, T(1)=1.

$$T(n) = 2\,T\left(\frac{n}{5}\right) + n$$

$$T\left(\frac{n}{5}\right) = 2T\left(\frac{n}{25}\right) + \frac{n}{5}$$

$$\Rightarrow T(n) = 2\left[2T\left(\frac{n}{25}\right) + \frac{n}{5}\right] + n = 4T\left(\frac{n}{5^2}\right) + \left(\frac{2n}{5} + n\right)$$

$$T\left(\frac{n}{25}\right) = 2T\left(\frac{n}{5^3}\right) + \frac{n}{25}$$

$$\Rightarrow T(n) = 4\left[2T\left(\frac{n}{5^3}\right) + \frac{n}{5^2}\right] + \left(\frac{2n}{5} + n\right) = 2^3 T\left(\frac{n}{5^3}\right) + \left(\frac{2^2}{5^2} + \frac{2}{5} + 1\right)n$$

$$\Rightarrow T(n) = 2^k \cdot T\left(\frac{n}{5^k}\right) + \left(1 + \frac{2}{5} + \left(\frac{2}{5}\right)^2 + \cdots + \left(\frac{2}{5}\right)^{k-1}\right) n$$

for $\frac{n}{5^k} = 1 \Rightarrow 5^k = n \Rightarrow k = \log_5 n$

$$\Rightarrow T(n) = 2^{\log_5 n} \cdot T(1) + \left[1 + \left(\frac{2}{5}\right) + \cdots + \left(\frac{2}{5}\right)^{\log_5 n - 1}\right] \cdot n$$

as $T(1)=1 \Rightarrow$

$$T(n) = 2^{\log_5 n} + \frac{1 \cdot \left[1 - \left(\frac{2}{5}\right)^{\log_5 n}\right]}{\left(1 - \frac{2}{5}\right)} \cdot n$$

$\because$ no of elements in G.P = $\log_5 n$

Sum = $\frac{a \cdot (1 - r^n)}{(1 - r)}$ $\because r < 1$

6

$$= 2^{\log_5 n} + \frac{5}{3}\left(1 - \frac{2^{\log_5 n}}{n}\right) \cdot n = 2^{\log_5 n} + \frac{5}{3}\left(n - 2^{\log_5 n}\right)$$

$$\boxed{T(n) = \frac{5}{3} n - \frac{2}{3} 2^{\log_5 n} = \frac{1}{3}\left[5n - 2^{1 + \log_5 n}\right]}$$

7. [8] Max-heap is a heap where the value at a node is larger than the values at the children. Given an array representation of a max-heap, complete the following code for **heapify** operation.

```
void heapify(int A[], int m)  { // m is the number of elements in A
   int i, n=m/2; int k;
   for ( i=n; i>=0;i--){
      if (A[i] ≱ A[2i] && A[i] ≥ A[2i+1])
      {   ~~k = 2A[2i]~~; k=A[i];
          A[i]= A[2i];   A[2i]= k;
          percolate_down(A, 2i);
      }
      else if ( A[i] < A[2i+1] && A[2i+1] > A[2i])
      {
          k = A[i]
          A[i]= A[2i+1];
          A[2i+1]= k;
          percolate_down(A, 2i+1);
      }
   }

void percolatedown (int A[], int k)
{ int l; if (2k>=N) return;
  else if (A[k] < A[2k] && A[k] ≥ A[2k+1])
  {   l = A[k];
      A[k]= A[2k];  A[2k]= l;
      percolatedown(A, 2k);
  }
  else if (A[k] < A[2k+1] && A[2k+1] > A[2k])
  {   l = A[k];  A[k] = A[2k+1];  A[2k+1]= l;  percolate_down(A,
}
```

7

Check if children exist or not? right child?