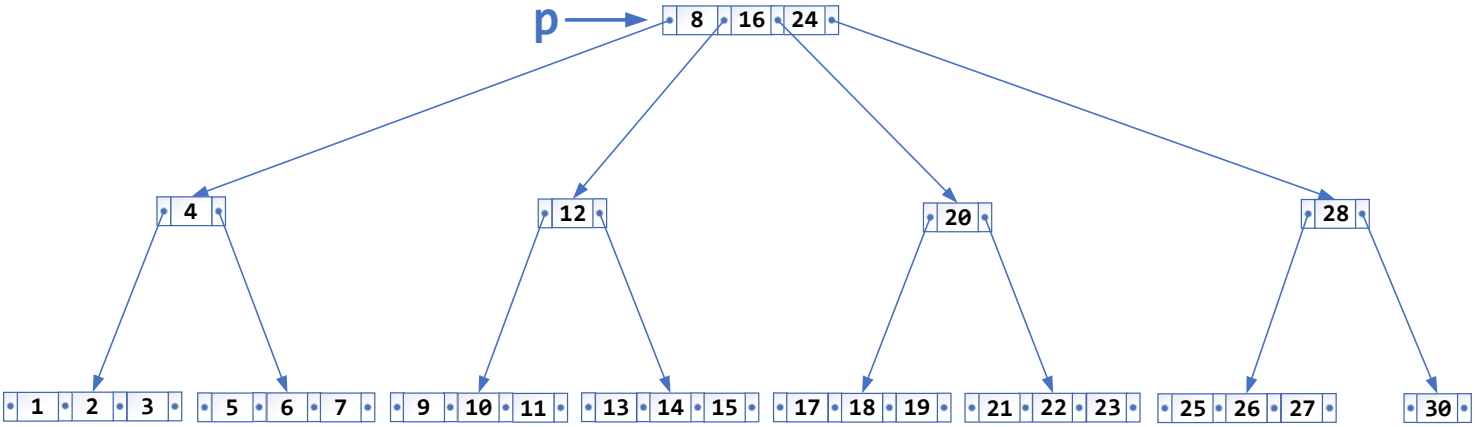
Karadeniz Technical University
Department of Computer Engineering
Lecturer Ömer ÇAKIR

COM 2005 Data Structures
Resit Exam, 05.02.2024, 15:00
Duration : **105** Minutes

SOLUTIONS

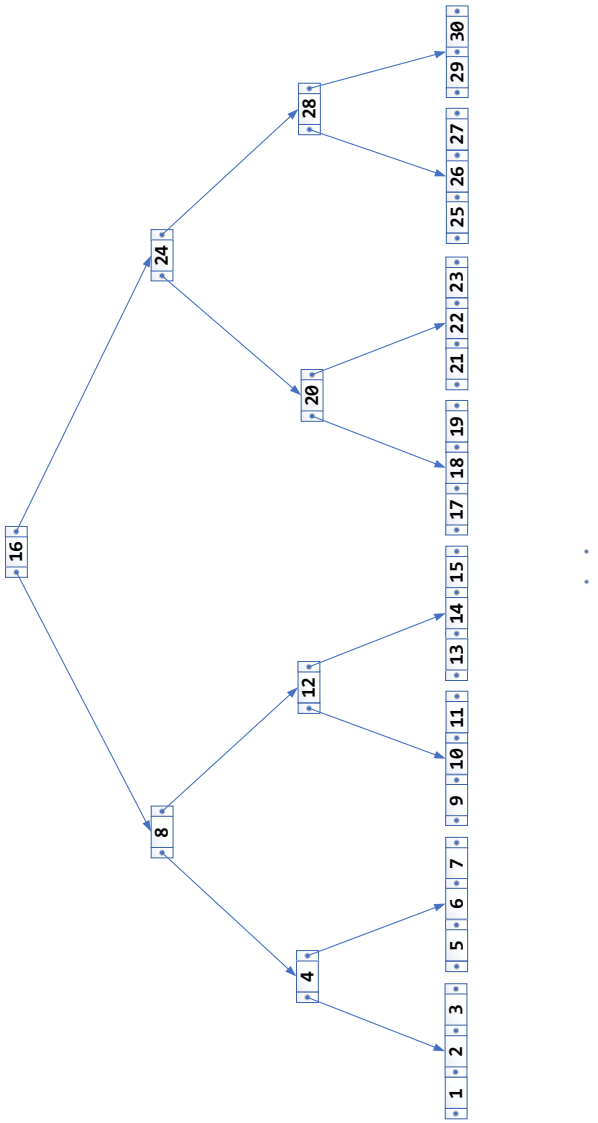



```
DoublyLinkedList* pLeft = new DoublyLinkedList;
pLeft->insertOrdered(p->hNextScr(1));

DoublyLinkedList* pRight = new DoublyLinkedList;
pRight->insertOrdered(p->hNextScr(3));

if (p->downPtr(1)->down != NULL)
{
  pLeft->downPtr(1)->down = p->downPtr(1)->down;
  pLeft->downPtr(1)->down->up = pLeft->downPtr(1);
}

if (p->downPtr(2)->down != NULL)
{
  pLeft->downPtr(2)->down = p->downPtr(2)->down;
  pLeft->downPtr(2)->down->up = pLeft->downPtr(2);
}

if (p->downPtr(3)->down != NULL)
{
  pRight->downPtr(1)->down = p->downPtr(3)->down;
  pRight->downPtr(1)->down->up =pRight->downPtr(1);
}

if (p->downPtr(4)->down != NULL)
{
  pRight->downPtr(2)->down = p->downPtr(4)->down;
  pRight->downPtr(2)->down->up =pRight->downPtr(2);
}

DoublyLinkedList* pUpper = new DoublyLinkedList;
pUpper->insertOrdered(p->hNextScr(2));
root = pUpper;

pUpper->downPtr(1)->down = pLeft;
pLeft->up = pUpper->downPtr(1);

pUpper->downPtr(2)->down = pRight;
pRight->up = pUpper->downPtr(2);
```

1. Assuming **29** is inserted into the **2-3-4 tree** above, which node does **pLeft** point to after insertion? **(20P)**

8

2. Draw the **2-3-4 tree** after inserting **29**. **(20P)**



```
int binarySum(int A[], int i, int n)
{
   if (n == 1)
      return A[i];
   else
   {
      int Sum = binarySum(A, i, n / 2) +
                binarySum(A, i + n / 2, n / 2);
      cout << Sum << " ";
      return Sum;
   }
}

int main()
{
  int A[16]= { 0,1,0,3,0,5,0,7,0,9,0,11,0,13,0,15};
  int binSum = binarySum(A, 0, 16);
}
```

3. What is the output of the program above? **(20P)**

1	3	4	5	7	12	16	9	11	20	13	15	28	48	64

4. How many times does the function **binarySum()** call itself recursively? **(20P)**

30

```
int main()
{
      CircularlyLinkedQueue Queue;

      Queue.enqueue(1);
      Queue.enqueue(4);
      Queue.enqueue(6);
      Queue.enqueue(1);

      Queue.dequeue();
      Queue.dequeue();
      Queue.dequeue();

      Queue.enqueue(9);
      Queue.enqueue(6);
      Queue.enqueue(7);

      Queue.C.print();
}
```

5. What is the output of the program above? **(20P)**

1
9
6
7