CS 2420
Public Exam 1

Multiple Choice (3 points each) Pick the single best answer

1. For the timing information below, what is the complexity?

n	T(n)
2	9
4	38
8	153
16	614
32	2457

(a) O(1) (b) O(log n) (c) O(n) (d) O(n log n) (e) O(n^2) (f) can't determine

2. From our theorem we know:
T(n)= a T(n/b) + O(n^k)
if a > b^k T(n) is O(n^{log_b a})
if a = b^k T(n) is O(n^k{log n})
if a < b^k T(n) is O(n^k)
Consider the following algorithm:

```
int doit( int n){
   if (n <=1) return 1;
   int t=0;
   for (int i =0; i < n; i++)
      t++;
   return doit(n/2)+ doit(n/2) + t;
}
```

What is the complexity?
(a) O(1) (b) O(log n) (c) O(n) (d) O(n log n) (e) O(n^2) (f) theorem does not apply

3. For the problem above, what are the values of a, b and k?

   a. a=1, b=2, k=0
   b. a=1, b=2, k=2
   c. a=2, b=2, k=0
   d. a=2, b=2, k=1
   e. none of the above

4. An algorithm has complexity O(n log n). What do you expect to happen to the execution time if the problem size doubles?

a. doubles
b. slightly more than doubles
c. quadruples
d. increases by a constant
e. stays the same

5. When a method call is executed, which information is not saved in the activation record?
   a. Current depth of recursion.
   b. Formal parameters.
   c. Location where the method should return when done.
   d. Local variables.

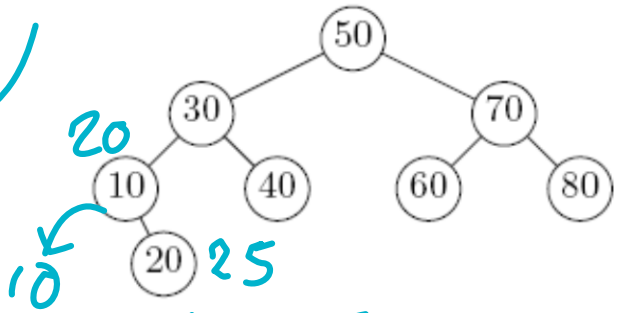
6. Consider this method declaration:

```
void quiz(int i)
{
  if (i > 1)
  { quiz(i / 2);
    quiz(i / 2);
  }
  System.out.print("*");
}
```

How many asterisks are printed by the method call quiz(5)? Draw a picture to illustrate what happens in the call.

a. 3 b. 4 c. 7 d. 8 e. some other number.

7. Suppose that we insert a 25 into the AVL tree below. What rotation would be used to fix the balance, according to the algorithm we discussed in class? Show the resultant tree.



A. A single rotation rooted at the node containing 10.
B. A single rotation rooted at the node containing 30.
C. A single rotation rooted at the node containing 50.
D. A double rotation rooted at the node containing 10.
E. A double rotation rooted at the node containing 50.

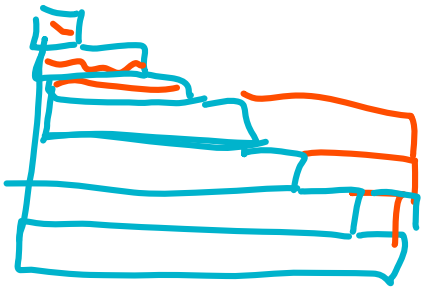
Short Answer

1. (5 points) What is the complexity of the following piece of code? Draw an appropriate picture to justify your answer.


```
void doit(int n)
{ if (n/2 <=1) return;
  doit(n/2);
  for (int i = 0; i< n;i++) cout <i;
  doit(n/2);
}
```

n^2 or $n \log n$

2. (5 points) What is the complexity of the following piece of code? Draw an appropriate picture to justify your answer.



```
void doit(int n)
{ int it;
  for (it=0;it<n;it++)
    for (int j=0;j<it;j++)
      cout << it * j;
}
```

$n \log n$

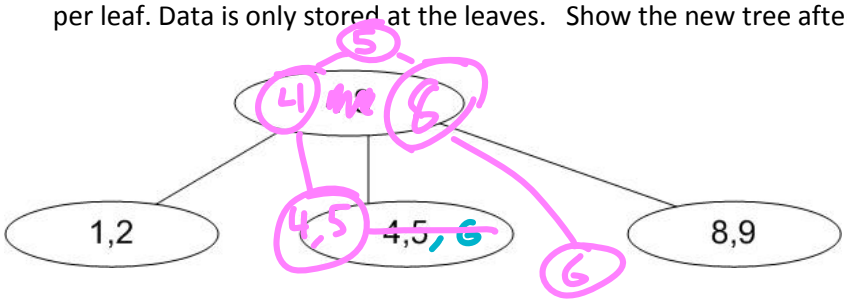
3. (10 points) You wrote a recursive routine to count the number of zeroes in the array. Your attempt and its output is shown below. Explain what the problem is and fix it.

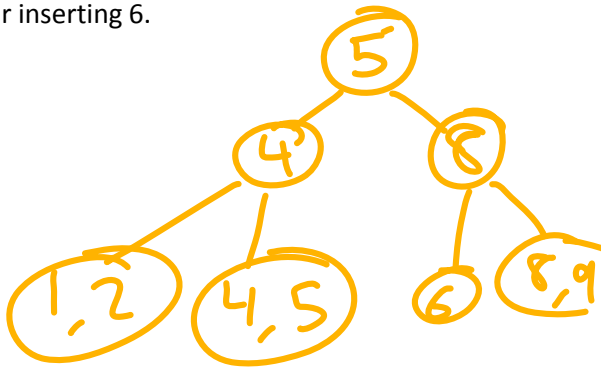
```
int any(int a[], int low, int high)
{ if (low>high) return 0;
  if (low==high) {
        if (a[low]==0) return 1;
      return 0;
  }
  int mid = (low + high)/2;
  return any(a,low,mid) + any(a,mid,high);
}

void main ()
{int a[16] = {0, 6, 0, 23, 0, 25, 50, 80, 2, 5, 7, 14, 35, 36, 37, 40};
 int m= any(a,0,15);
 if (m>0) cout << m << " zeroes\n";
 else cout << "NO zeros\n";
}
```

Output is:
(Infinite recursion)

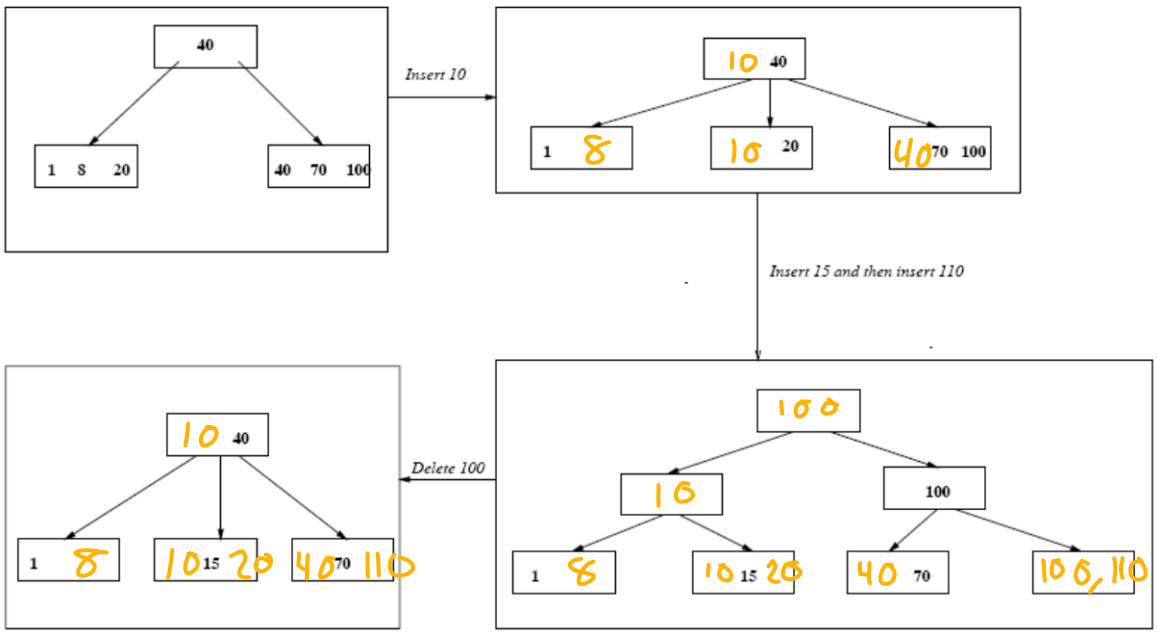
high will always be > mid
We need to do mid+1 to make this correct

4. **. (5 points) Consider the following B+ tree.** A new key 6 is to be inserted into the following B-tree. This B+-tree is allowed to have two to three children per internal node and one or two records per leaf. Data is only stored at the leaves. Show the new tree after inserting 6.

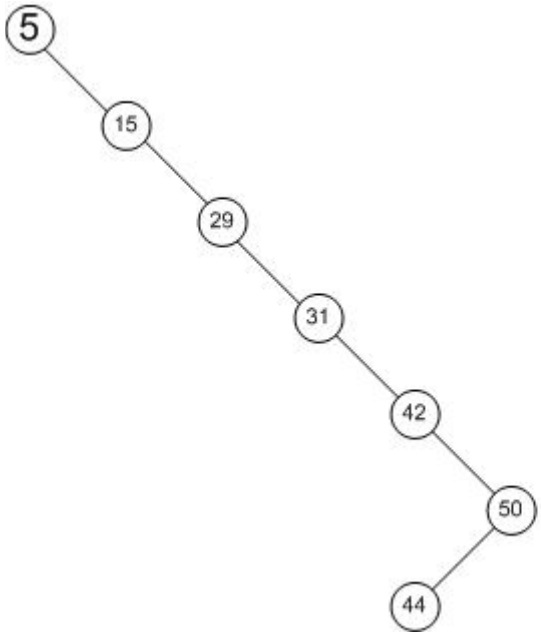




5. (10 points) Consider a B+-tree that has two to three children per internal node and two or three records per leaf. Data is only stored at the leaves. The figure below depicts various stages in the life of such a tree. Each figure has one or more keys missing. Complete the figures based on your knowledge of how the B+-tree would be restructured after each insertion and deletion.



6. (6 points) Given a top down splay tree below, show the result after a splay on 50.



50
42
31
44
29
15
5

7. 10. (4 points) What is the complexity of doing a find in an AVL tree? What property of an AVL tree is most significant in explaining their Big Oh behavior for find?

log(n), it's always balanced

8. (14 points) Write the code to verify that a tree is an AVL tree. Note this means you have to insure that it is balanced and that it is a binary search tree. Assume the following node structure.

```
class AVLNode
{
  public:
    char element;
    AVLNode *left;
    AVLNode *right;
    AVLNode(char e = ' ', AVLNode *l = NULL, AVLNode *r = NULL ){
     element=e; left =l; right = r;  }

};
```