**1、Which case is most commonly used to evaluate the running time of one algorithm.** ( )

(A) Worst case (B) Average case (C) Best case (D) Ideal case.

**2、Which method is not used to resolve a recurrence.** ( )

(A) Substitution method (B) Recursion tree method

(C) Master method (D) Linear programming method.

**3、**An order-statistic tree is an augmented red-black tree. In addition to its usual fields, each node x has a field size[x], which is the number of nodes in the subtree rooted at x. For an order-statistic tree with *n* nodes, the time for insertion, deletion and maintenance of the *size* field are ( )

(A)    (B)   

(C)    (D)   

**4、**Computing a discrete Fourier transform (DFT) of *N* points takes \_\_ arithmetical operations, while computing a fast Fourier transform (FFT) of *N* points takes \_\_ arithmetical operations. ( )

(A)  (B)  

(C)   (D)  

**5、**In a hash table in which collisions are resolved by chaining, a successful search takes time \_\_, under the assumption of simple uniform hashing. The load factor of this hash table is . ( )

(A) (B)  (C)  (D) 

**6、**Which of the following sorting algorithms is not stable? ( )

(A) Insertion sort (B) Quick sort

(C) Merge sort (D) Bubble sort

**7、**We say that is asymptotically larger thanif ( ).

(A)  (B) 

(C)  (D) 

**8、**There’s a B-tree whose minimum degree is t, every node other than the root must have at least \_\_ keys, at most \_\_ keys, every internal node other than the root has at least \_\_ children ( ).

(A) t-1 2t t (B) t-1 2t-1 t

(C) t 2t t+1 (D) t-1 2t+1 t

**1.Please write down the red-black properties. [5 points]**

If a node is red, then both its children are black.

For each node, all paths from the node to descendant leaves contain the same number of black nodes.

**2.What is the largest possible number of internal nodes in a red-black tree with black-height k?**

**What is the smallest possible number? [3 points]**

Consider a red-black tree with black-height k. If every node is black the total number of internal nodes is 2k - 1.

If only every other nodes is black we can construct a tree with 22k - 1 nodes.

**1.Please write down the elements of dynamic programming.**

**(1) Optimal substructure**

A problem exhibits Optimal substructure if an optimal solution to the problem is contained within its optimal solutions to subproblems.

**(2)Overlapping subproblems**

When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has Overlapping subroblems

**2.Tell the difference between dynamic programming and greedy programming. [6 points]**

（1）They share the optimal substructure property, but we may not use dynamic programming when a greedy solution suffices, or reverse.

（2）The greedy choice property is that a globally optimal solution can be arrived at by making a locally optimal (greedy) choice.

（3）We must prove that a greedy choice at each step yields a globally optimal solution.

First consider a globally optimal solution, and then modify the optimal solution, to make it to begin as a greedy choice.

**递归树**

1.Using a recursion tree to give an asymptotically tight solution to the recurrence *T*(*n*) = *T(n/3)+T(2n/3)+cn.* [9 points]

2.Using a recursion tree to give an asymptotically tight solution to the recurrence *T*(*n*) = *T(n/4)+T(n/2)+ n2.*

3.用主方法来给出下列递归式的渐近界：

a) 

b) 

c) 

4.Use a recursion tree to give an asymptotically tight solution to the recurrence *T*(*n*) = *T*(*αn*) + *T*((1 - *α*)*n*) + *cn*, where *α* is a constant in the range 0 <α < 1 and *c* > 0 is also a constant. [12 points]

**排序**

1.Using figure to illustrate the operation of COUNTING-SORT on the array

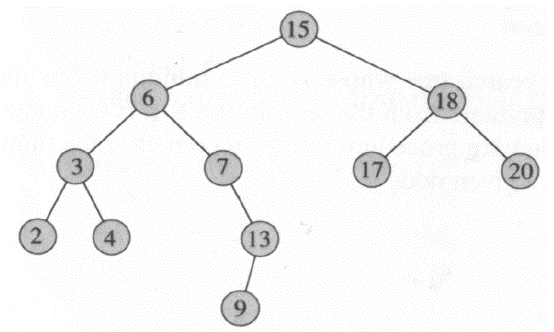
A=<6,0,2,0,1,3,4,6,1,3,2>.

2.Using figure to illustrate the operation of RADIX-SORT on the following list of English words:

COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB. [8 points]

**遍历**

Please write inorder, preorder and postorder tree walks of the following binary search tree. [9 points]



**Hash表**

1. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length m = 11 using open addressing with the primary hash function h1(k) = k mod m. Illustrate the result of inserting these keys using linear probing, using quadratic probing with c1 = 1 and c2 = 3. The final results are required to be expressed as charts. [6 points]

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 22 | 88 |  |  | 4 | 15 | 28 | 17 | 59 | 31 | 10 |

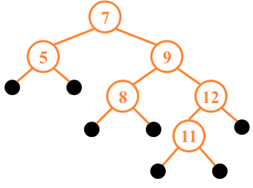
**红黑树**

11、A red-black tree is a binary search tree with one extra bit of storage per node: its color, which can be either RED or BLACK, and the red-black is a nearly balanced tree. [10 points]

1)Please prove the lemma: A red-black tree with n internal codes has height at most ?

2)Please prove n-node RB tree has height 

2.Please draw the result after the operation Left-Rotate(9)

.

1. Put the keywords 41,83,31,12,19, 8 into an initially empty red-black tree, then draw the result tree
2. In turn, given from the first question of red-black tree, delete keywords 8,12,19,31,83,41 .

**动态规划**

1.X=<A, E, B, D, B, C, A, E>, Y=<E, F, B, A, C, A, F, E>. Please illustrate the whole procedure for finding the longest common sequence of X and Y using dynamic programming.

2.Suppose A1 a  matrix, A2 a  matrix, A3 a  matrix, A4 a  matrix, A5 a  matrix, A6 a  matrix. Please give an optimal parenthesization of a matrix-chain A1A2A3A4A5A6.

Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is

<5, 10, 3, 12, 5, 50, 6 >. [10 points]

**线性规划**

1.Converting the following linear program into standard form:

Minimize 

Subject to 







2.Solve the following linear program using SIMPLEX:

maximize 

Subject to 







**贪婪算法**

1.Please give an optimal Huffman code for the following set of frequencies.

|  |
| --- |
| a b c d e f |
| Frequency 5 9 16 12 13 45 |

1.Please give an optimal Huffman code for the following set of frequencies.

|  |
| --- |
| a b c d e f |
| Frequency 15 19 6 12 13 35 |

2.In the activity-selection problem, represents the activities that start after an activity  finishes and finish before one activity  starts. Here, an activity  occurs during period, and activities are sorted by monotonically increasing finish time. Let, and let  be the activity in with the earliest finish time:. Then prove  is used in some maximum-size subset of mutually compatible activities of 