四川大学期中考试试题（闭卷）

（2023~2024 学年第 1 学期）

A 卷

课程号-课序号：311076040　课程名称：数据结构与算法　任课教师：李晓华

适用专业年级：软件工程 等 2022 级　学生人数：156　印题份数：160　学号：[redacted]　姓名：[redacted]

考生承诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学生考试违纪作弊处分规定（修订）》，郑重承诺：

1. 已按要求将考试禁止携带的文具用品或与考试有关的物品放置在指定地点；
2. 不带手机进入考场；
3. 考试期间遵守以上两项规定。若有违规行为，同意按照有关条款接受处理。

考生签名：

题　号	一 (30%)	二 (45%)	三 (25%)
得　分			
卷面总分		阅卷时间	

注意事项：1. 请务必将本人所在学院、姓名、学号、任课教师姓名等信息准确填写在试题纸和添卷纸上；

2. **请将答案全部填写在答题纸上；本试题纸上的答案一律不计分；**

3. 考试结束，请将试题纸、添卷纸和草稿纸一并交给监考老师。

评阅教师	得分

一、单项选择题（本大题共 10 小题，每小题 3 分，共 30 分）

提示：在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在答题纸上。错选、多选或未选均无分。

1. If R is a binary relation over set S, then R is reflexive if (B)
   A. aRa for all a in S.
   B. whenever aRb, then bRa, for all a, b in S.
   C. whenever aRb and bRa, then a = b, for all a, b in S.
   D. whenever aRb and aRc, then aRc, for all a, b, c in S.

2. If the preorder of a binary tree is ABCDEFG, then the possible inorder is (A).
   A. ABCDEFG　B. CABDEFG　C. DACEFBG　D. ADCFEGB

3. Consider the node of a complete binary tree whose value is stored in data[i] for an array implementation. If this node has a left child, where will the right child's value be stored? (C)
   A. data[i+1]　B. data[i+2]　C. data[2*i + 1]　D. data[2*i + 2]

4. If f(n) = O(g(n)) and f(n) = Ω(g(n)), then it is always true that (C)
   A. f(n) = O(g(n)).　B. f(n) = Θ(g(n)).　C. f(n) = ω(g(n)).　D. both a) and b) are always true.

5. What is the worst case time complexity for search in a general tree? (C)
   A. n　B. nlogn　C. n^2　D. None of above

注：试题字迹务必清晰，书写工整。

6. In the binary search, the linear list is required (C).
   A. array structure.　B. array structure or linked list structure.
   C. array structure and the elements are sorted.　D. linked list structure and the elements are sorted.

7. If the MaxSize of a circular queue is 6, rear points to the 0th element and front points to the 3rd element in the queue. After removing two elements and inserting one element, rear and front point to the (A) elements respectively.
   A. 1st and 5th　B. 2nd and 4th　C. 4th and 2nd　D. 5th and 1st

8. Which structure is convenient for dynamic inserting and deleting (B).
   A. array　B. link list　C. stack　D. queue

9. For a list of length n, the linked-list implementation's prev function requires worst-case time: (C)
   A. Θ (1).　B. Θ (log n).　C. Θ (n).　D. Θ (n^2).

10. The time required to access and insert an element in an array-based list respectively is (D)
    A. Θ(n) Θ(n)　B. Θ(n) Θ(1)　C. Θ(1) Θ(n)　D. Θ(1) Θ(1)

评阅教师	得分

二、应用题（本大题共 4 小题，每小题 10-12 分，共 45 分）

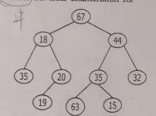
提示：有求解过程的要尽量给出解题步骤，只有最终答案会酌情扣分。

1. (11 分) Starting from an empty binary tree, sequentially insert the following elements one-by-one according to the insertion algorithm of binary search tree: 23,10,35,28,40,45,42,53.
   (a) Draw the binary search tree after inserting all the above elements.
   (b) Draw the binary search tree after deleting the element with value 35.

2. (11 分) Please show the preorder traversal enumeration and inorder traversal enumeration for the following tree.

3. (12 分) Given the following set of letters and weights:

A	B	C	D	E	F	J	H
3	5	6	8	12	19	23	22



   (a) Building the Huffman coding tree
   (b) What is the best-case number of bits required by the Huffman code for a message containing 100 characters?
   (c) What is the expected number of bits required by the Huffman code for a message containing 10000 characters

4. (11 分) (a) Show the max-heap on the following values stored in an array:
   25, 32, 19, 37, 39, 61, 10,33, 65, 27.
   (b) Show the heap that results from deleting the maximum value from the max-heap of (a).

注：试题字迹务必清晰，书写工整。

评阅教师	得分

三、程序设计题（本大题共2小题，1小题10分，2小题15分，共25分）。

提示：下面给出了一个程序设计要求，请按照要求写出源程序代码，如果源程序代码中出现语法错误或逻辑错误，则酌情扣分。

1. Given a non-empty queue Q, and an empty stack S, using only stack and queue ADT functions, and a single element variable X, write an algorithm to reverse the order of the elements in Q.

2. Given an array storing integers ordered by value, modify the following binary search routine to return the position of the first integer with value K in the situation where K can appear multiple times in the array. Be sure that your algorithm is in O(log n), that is, do not resort the sequential search once an occurrence of K is found.

```
int  binary(int K, int A[], int n)
{
   int  l = -1;
   int  r = n;
   while ( (l+1) != r)
   {
      int  i = (l+r)/2;
      if (K < array[i])      r = i;
      if (K == array[i]) return i;
      if (K > array[i])      l = i;
   }
   return n;
}
```