

四川大学期末考试试题（闭卷）

（2017~2018 学年第 1 学期）

A 卷

课程号: 311076040 课程名称: 数据结构与算法 任课教师: _____

适用专业年级: 软件工程 2016 级 学号: _____ 姓名: _____

考生承诺

我已认真阅读并知晓《四川大学考场规则》和《四川大学本科学子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(16%)	三(34%)	四(20%)
得 分				
卷面总分		教师签名	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 15 小题，每小题 2 分，共 30 分）

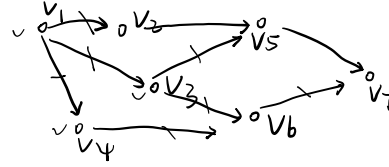
提示：在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码写在答题纸上。错选、多选或未选均无分。

- ① The most critical factor which is used to estimate an algorithm's performance is (~~D~~) C
- (A) Easy to debug
- (B) Easy to understand
- (C) Time complexity of an algorithm 度量算法性能
- (D) Correctness of an algorithm
2. Given a single linked list with head node, (B) means that the list is empty.
- (A) HEAD = NULL
- (B) HEAD->NEXT = NULL
- (C) HEAD->NEXT = HEAD
- (D) HEAD != NULL
3. Assume that a pointer top is set to point to the first link node of a stack, then the operation of removing the top node is (D).
- (A) top = top+1;
- (B) top = top-1;

- (C) $\text{top} \rightarrow \text{next} = \text{top};$
 (D) $\text{top} = \text{top} \rightarrow \text{next};$
4. A BST conforms to the following condition: All nodes stored in the left subtree of a node whose key value is K have key values (A) K .
- (A) $<$
 (B) $>$
 (C) $=$
 (D) \neq
5. If the number of leaves for a Huffman tree is n then the nodes number of this Huffman tree is (D).
- (A) uncertainty $n_0 = n - 1$
 (B) $2n$
 (C) $2n + 1$
 (D) $2n - 1$ $n = (n - 1) + n = 2n - 1$
6. If the sequence $\{11, 12, 13, 7, 8, 9, 23, 4, 5\}$ is the middle result after one sorting pass, then the sort method used is (B).
- (A) Bubble sort
 (B) Insertion sort
 (C) Selection sort
 (D) Two-way Mergesort
7. If we sort all people in China by the date of birth (Month and Day), which sorting algorithm is the fastest? (D) A
- (A) Radix sort
 (B) Mergesort
 (C) Heapsort
 (D) Quicksort
8. A perfect hash function can store the actual records in the collection such that each slot in the hash table has (D) of being filled.
- (A) the maximum probability
 (B) the minimum probability
 (C) average probability
 (D) equal probability
9. A B-tree of order m is defined to have the following shape properties except (D).
- (A) The root is either a leaf or has at least two children.
 (B) All leaves are at the same level in the tree.
 (C) Each internal node, except for the root, has between $\lceil m/2 \rceil$ and m children.

(D) The leaf nodes are linked together to form a doubly linked list.

10. Given a directed graph $G=(V, E)$, $V=\{V1, V2, V3, V4, V5, V6, V7\}$, $E=\{<V1, V2>, <V1, V3>, <V1, V4>, <V2, V5>, <V3, V5>, <V3, V6>, <V4, V6>, <V5, V7>, <V6, V7>\}$, then the possible sequence of topological sort is (A)



- (A) V1, V3, V4, V6, V2, V5, V7
- (B) V1, V3, V2, V6, V4, V5, V7
- (C) V1, V3, V4, V5, V2, V6, V7
- (D) V1, V2, V5, V3, V4, V6, V7

11. Given the input order of a stack is A, B, C, D, E, F, (C) is not the possible valid output order.

- (A) B C D A F E
- (B) C B D F E A
- (C) D C A B E F
- (D) E D C F B A

12. In the following items, (C) is not correct about liner structure.

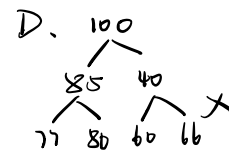
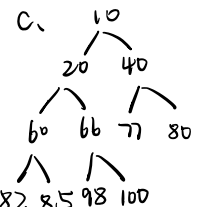
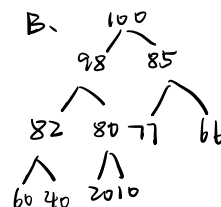
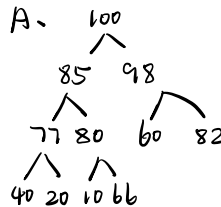
- (A) A collection of items ordered by a single property
- (B) Linked representations are normally called lists
- (C) Has a unique "predecessor" and a unique "successor"
- (D) sequential representations are called arrays

A 有序线性表

ps=老师说选C

13. In the following sequence, (D) is not a heap.

- (A) 100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66
- (B) 100, 98, 85, 82, 80, 77, 66, 60, 40, 20, 10
- (C) 10, 20, 40, 60, 66, 77, 80, 82, 85, 98, 100
- (D) 100, 85, 40, 77, 80, 60, 66, 98, 82, 10, 20



14. The primary difference between a B-tree and a B+-tree is (A)

- (A) The B+-tree store records only at the leaf nodes.
- (B) The B+-tree has a higher branching factor.
- (C) The B+-tree is hight balanced.
- (D) The B+-tree is smaller.

15. The basic unit of I/O when accessing a disk drive is (B)

- (A) A byte
- (B) A sector → Block
- (C) A cluster
- (D) An extent

从物理结构 = sector

从OS角度 = cluster

评阅教师	得分

二、名词解释题（本大题共 4 小题，每小题 4 分，共 16 分）。

提示：解释每小题所给名词的含义，若解释正确则给分，若解释错误则无分，若解释不准确或不全面，则酌情扣分。

(已不老)

1. DAG
2. Shell Sort
3. Huffman Coding
4. Buffer Pool

评阅教师	得分

三、应用题（本大题共 4 小题，1-2 每小题 8 分，3-4 每小题 9 分，共 34 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. Suppose you have a binary tree whose data fields are single characters. When the nodes are output in in-order traversal, the output is DCEFBHGAKJLIM, and when they are output in post-order, the output is DFECHGBKLIJMA. Draw the binary tree showing the data in each node.

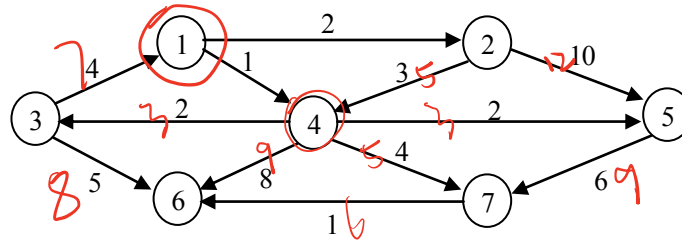
2. The following function is searching K from a sorted array with size of n.

- 1) Determine running time $T(n)$ in the best case, worse case and average case;
- 2) Determine Θ in the average case.

```
int Search (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;
}
```

3. Given following values stored in an array: 19, 64, 39, 56, 25, 47, 30, 73, 11, 67, 89.
 - (a) Construct the max-heap of these values.
 - (b) Show the heap that results from deleting the maximum value from the max-heap of (a).
4. Show the shortest paths generated by running Dijkstra's shortest-paths algorithm on the following

graph, beginning at Vertex 1. Show the D values as each vertex is processed.



评阅教师	得分

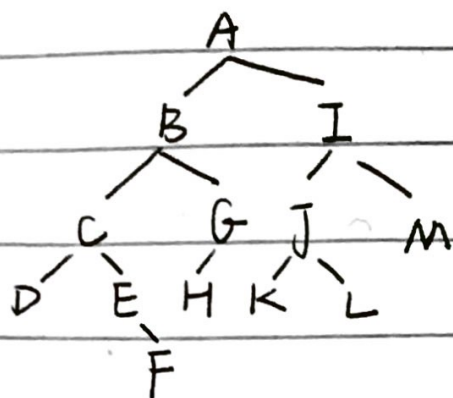
四、编程、设计及分析题（本大题共 2 小题，1 小题 8 分，2 小题 12 分，共 20 分）。

提示：每小题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. Write a function to compute the max width of a binary tree. (max width means the maximum value of each level 二叉树所有层中结点个数的最大值). (8 points)
2. A sorting algorithm named Count Sorting. It sorted an unordered array-based sequence to a new array, please pay attention that the keys to be sorted are different from each other. Count Sorting Algorithm scanned each record in array, counted the number of keys which is smaller than current key. Suppose that the counted count value of a key is C, the proper storage location of this record in the new ordered array is C.

Write the algorithm of Count Sorting. (12 points)

三、1. Binomy Tree:



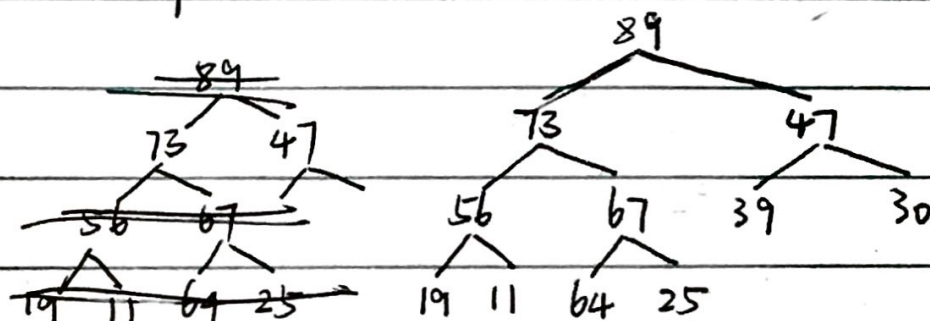
2. (1) best case: $T(n)=1$

worst case: $T(n)=\log_2 n$

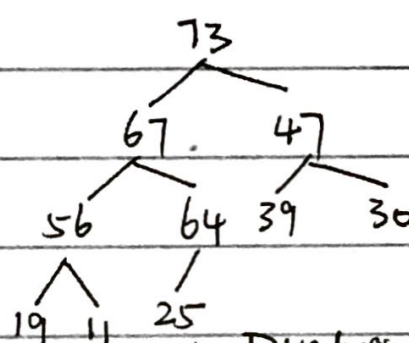
average case: $T(n)=\log_2 n$

(2) $\Theta(\log_2 n)$

3. (a) max-heap:



(b) 删除最大值 89 后:



4. Shortest path:

最终 PE 值

	2	3	4	5	6	7
P	1	4	1	4	4	4

由 $V_2 = V_2 \leftarrow V_1$ 到 $V_6 = V_6 \leftarrow V_4 \leftarrow V_1$

由 $V_3 = V_3 \leftarrow V_4 \leftarrow V_1$ 到 $V_7 = V_7 \leftarrow V_4 \leftarrow V_1$

由 $V_4 = V_4 \leftarrow V_1$

由 $V_5 = V_5 \leftarrow V_4 \leftarrow V_1$

D values:

	2	3	4	5	6	7
初始	2	∞	①	∞	∞	∞
处理4	②	3	1	3	9	5
处理2	2	③	1	3	9	5
处理3	2	3	1	③	8	5
处理5	2	3	1	3	第8 ⑤	5
处理7	2	3	1	3	⑥	5
处理6	2	3	1	3	6	5

四. 11 计算树的高度

```
1. int heightTree (const Node* subroot) {
    if (subroot == NULL) return;
    int leftH = heightTree (subroot->leftChild);
    int rightH = heightTree (subroot->rightChild);
    return (leftH > rightH ? (leftH + 1) : (rightH + 1));
}
```

11 计算第 ^{depth} ~~level~~ 层宽度 ~~width~~ ^{int & rwidth}

```
void WidthLevel (const Node* subroot, const int depth, int currDepth,
    if (subroot == NULL || currDepth > depth) return;
    else if (currDepth < depth) {
        WidthLevel (subroot->leftChild, depth, currDepth + 1, width);
        WidthLevel (subroot->rightChild, depth, currDepth + 1, width);
    }
    else if (currDepth == depth) { // 到达第 depth 层, 开始计数该层结点数
        width++;
    }
}
```

```
void int MaxHeight (const Node* root) {
    int height = heightTree (root);
    int width = 0, maxWidth = 0, depth = 0;
    for (int i = 0; i < height; i++) { // 依次计算 0 ~ height-1 层宽度, 求 max.
        width = 0; // 初始化该层宽度, 先置 0
        WidthLevel (root, i, 0, width);
        if (width > maxWidth) maxWidth = width;
    }
    return maxWidth;
}
```

2. void swap(Elem A[], int i, int j){

Elem Temp = A[i];

A[i] = A[j];

A[j] = Temp;

}

void countSorting(Elem A[], int n){

for (int i=0; i<n; i++){

int count=0;

for (int j=0; j<n; j++){

if (A[j] < A[i]) count++;

swap(A, i, count);

}

}