2018-2019 学年第一学期期中模拟练习

一、判断题(每小题 3 分，共 15 分)

1. (　) In a binary search tree, the keys on the same level from left to right must be in sorted (non-decreasing) order. (3 分)

2. (　) If keys are pushed onto a stack in the order {1, 2, 3, 4, 5}, then it is impossible to obtain the output sequence {3, 4, 1, 2, 5}. (3 分)

3. (　) If there are less than 20 inversions in an integer array, the Quick Sort will be the best method among Quick Sort, Heap Sort and Insertion Sort. (3 分)

4. (　) NlogN2 and NlogN have the same speed of growth. (3 分)

5. (　) For a sequentially stored linear list of length N, the time complexities for query and insertion are O(1) and O(N), respectively. (3 分)

二、选择题(共 64 分)

1、The result of performing three DeleteMin operations in the min-heap {1,3,2,12,6,4,8,15,14,9,7,5,11,13,10} is: (5 分)

E. 4,5,6,7,8,9,10,11,12,13,14,15
F. 4,5,6,12,7,10,8,15,14,13,9,11
G. 4,6,5,13,7,10,8,15,14,12,9,11
H. 4,6,5,12,7,10,8,15,14,9,13,11

2、For an in-order threaded binary tree, if the pre-order and in-order traversal sequences are B E A C F D and A E C B D F respectively, which pair of nodes' right links are both threads? (4 分)

A. E and F

B. B and E

C. A and D

D. A and E

3、Among the following sorting methods, which ones will be slowed down if we store the elements in a linked structure instead of a sequential structure? (5 分)

1. Insertion sort; 2. Selection Sort; 3. Bubble sort; 4. Shell sort; 5. Heap sort

E. 4 and 5 only

F. 1 and 2 only
G. 2 and 3 only

H. 3 and 4 only

4、Insert { 5, 11, 13, 1, 3, 6 } one by one into an initially empty binary search tree. The post-order traversal sequence of the resulting tree is: (6 分)

E. 6 is the root

F. 2 and 6 are siblings

G. 2 is the parent of 4

H. None of the above

5、To sort { 49, 38, 65, 97, 76, 13, 27, 50 } in increasing order, which of the following is the result after the 1st run of Shell sort with the initial increment 4? (5 分)

E. 13,27,38,49,50,65,76,97
F. 49,76,65,13,27,50,97,38

G. 49,13,27,50,76,38,65,97
H. 97,76,65,50,49,38,27,13

6、Given a quadtree(四叉树) with 2 nodes of degree 2, 3 nodes of degree 3, 4 nodes of degree 4. The number of leaf nodes in this tree is __. (5 分)

A. 12

B. 20

C. 21

D. 10

7、For the quicksort implementation with both the left and the right pointers stop when an element with the same key as the pivot is found during the partitioning, what is the running time when all keys are equal? (5 分)
E. O(NlogN)
F. O(N)
G. O(N2)
H. O(logN)

8、How many leaf node does a complete binary tree with 2435 nodes have? (5 分)

A. 1218

B. 812

C. cannot be determined

D. 1217

9、The recurrent equations for the time complexities of programs P1 and P2 are:

- P1: T(1)=1, T(N)=T(N/2)+1;
- P2: T(1)=1, T(N)=2T(N/2)+1;

Then the correct conclusion about their time complexities is: (5 分)
A. they are both O(N)

B. O(logN) for P1, and O(N) for P2

C. they are both O(logN)

D. O(logN) for P1, and O(NlogN) for P2

10、Insert { 5, 11, 13, 1, 3, 6 } one by one into an initially empty binary search tree. The post-order traversal sequence of the resulting tree is: (5 分)

A. 3, 1, 6, 13, 11, 5

B. 1, 3, 5, 6, 13, 11

C. 1, 3, 11, 6, 13, 5

D. 3, 1, 5, 6, 13, 11

11、Given input { 431, 56, 57, 46, 28, 7, 331, 33, 24, 63 }. Which one of the following is the result after the 1st run of the Least Signification Digit (LSD) radix sort? (5 分)

A. →57→46→28→7→33→24→63→56→431→331

B. →56→28→431→331→33→24→46→57→63→7

C. →331→431→33→63→24→56→46→57→7→28

D. →431→331→33→63→24→56→46→57→7→28

12、Suppose that an array of size **m** is used to store a circular queue. If the head pointer **front** and the current size variable **size** are used to represent the range of the queue instead of **front** and **rear**, then the maximum capacity of this queue can be: (5 分)

E. cannot be determined

F. m+1

G. m-1

H. m

13、To insert s after p in a doubly linked circular list, we must do: (6 分)

E. p->next=s; s->prior=p; p->next->prior=s ; s->next=p->next;

F. p->next->prior=s; p->next=s; s->prior=p; s->next=p->next;

G. s->prior=p; s->next=p->next; p->next=s; p->next->prior=s;

H. s->prior=p; s->next=p->next; p->next->prior=s; p->next=s;

三、程序填空题（满分 20 分）

1、(**9 分**) Modified Selection Sort

The function is to sort the list **{ r[1] … r[n] }** in non-decreasing order. Unlike selection sort which places only the minimum unsorted element in its correct position, this algorithm finds both the minimum and the maximum unsorted elements and places them into their final positions.

```
void    sort( list r[], int n )
{
    int i, j, mini, maxi;
    for (i=1; i<n-i+1; i++) {
        mini = maxi = i;
        for( j=i+1; _____(3 分); ++j ){
            if( _____(3 分)) mini = j;
            else if(r[j]->key > r[maxi]->key) maxi = j;
```

```
        }
        if( mini != i ) swap(&r[mini], &r[i]);
        if( maxi != n-i+1 ){
            if( __________(3 分) swap(&r[mini], &r[n-i+1]);
            else swap(&r[maxi], &r[n-i+1]);
        }
    }
}
```

2、（**6 分**）FindKthLarges.

The function is to find the K-th largest element in a list A of N elements. The function BuildMinHeap(H, K) is to arrange elements H[1] ... H[K] into a min-heap. Please complete the following program.

```
ElementType FindKthLargest ( int A[], int N, int K )
{    /* it is assumed that K<=N */
     ElementType *H;
     int i, next, child;
     H = (ElementType *)malloc((K+1)*sizeof(ElementType));
     for ( i=1; i<=K; i++ ) H[i] = A[i-1];
     BuildMinHeap(H, K);
     for ( next=K; next<N; next++ ) {
          H[0] = A[next];
          if ( H[0] > H[1] ) {
               for ( i=1; i*2<=K; i=child ) {
                    child = i*2;
                    if ( child!=K && __________(5 分) ) child++;
                    if (   ____________________(5 分) )
                         H[i] = H[child];
                    else break;
               }
               H[i] = H[0];
          }
     }
     return H[1];
}
```

四、函数题（满分 6 分）

No Greater Than X in BST

You are supposed to output, in decreasing order, all the elements no greater than X in a binary search tree T.

Format of function:

```
void Print_NGT( Tree T,    int X );
```

where `Tree` is defined as the following:

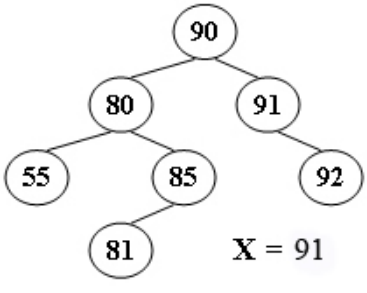
```
typedef struct TreeNode *Tree;
struct TreeNode {
    int Element;
    Tree    Left;
    Tree    Right;
};
```

The function is supposed to use `Output(X)` to print `X`.

Sample program of judge:

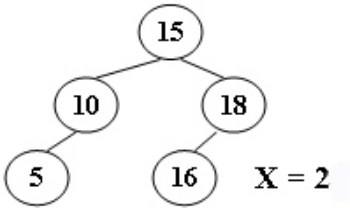
```
#include <stdio.h>
#include <stdlib.h>
typedef struct TreeNode *Tree;
struct TreeNode {
    int Element;
    Tree    Left;
    Tree    Right;
};
Tree BuildTree(); /* details omitted */
void Output( int X ); /* details omitted */

void Print_NGT( Tree T,    int X );
int main()
{
    Tree T;
    int X;
    T = BuildTree();
    scanf("%d", &X);
    Print_NGT( T, X );
    printf("End\n");
    return 0;
}
/* Your function will be put here */
```

Sample Output 1 (for the tree shown in Figure 1):




```
91 90 85 81 80 55 End
```

Sample Output 2 (for the tree shown in Figure 2):



2018-2019 学年第一学期期中模拟练习参考答案

一、判断题(共 15 分)

1、【正解】T

【解析】二叉搜索树中任意一个结点的值一定大于其左孩子的值而小于其右孩子的值（如果存在的话）。其中序遍历序列是非递减的，因此层序遍历也是非递减的。

【考点延伸】二叉搜索树

2、【正解】T

【解析】模拟出入栈，当 3，4 出栈后，栈中的元素序列为 1 2，此时 1 不可能在 2 之前出栈。

【考点延伸】栈

3、【正解】F

【解析】快速排序适合大量元素的排序，在元素数量较少时，插入排序的性能优于快速排序和对堆排序

【考点延伸】基本排序

4、【正解】T

【解析】前者可化为 2NlogN，可以将系数忽略，因此二者增长的速度相等

【考点延伸】时间复杂度

5、【正解】T

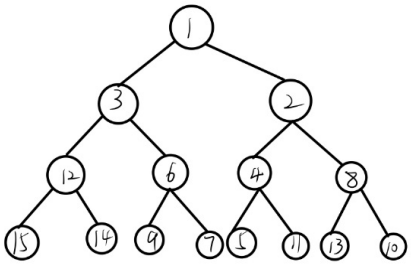
【解析】线性表支持随机访问，因此访问的时间复杂度为 O(1)。但是在第 i 个位置插入时，需将其后的 n-i 个元素分别向后移 1 位，因此时间复杂度为 O(n)。

【考点延伸】线性表

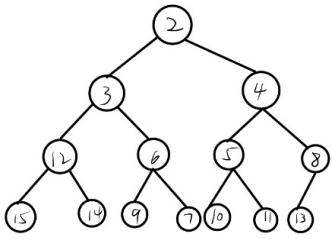
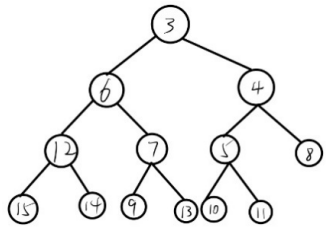
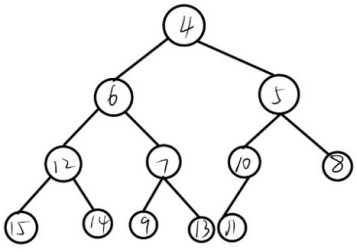
二、单选题(共 64 分)

1 【正解】D

【解析】最小堆如下图所示



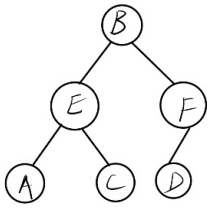
经过三次删除操作：

【考点延伸】最小堆

2 【正解】D

【解析】 二叉树如图所示



【考点延伸】二叉树的遍历

3 【正解】C

【解析】简单选择排序每一趟都需要扫描整个序列找到一个最值，和前面的交换。交换的时候无法利用顺序结构随机访问的特性，效率会变低。冒泡排序也是一样，在每一趟中，并不是每次都需要从头往后遍历。但使用链式结构后，无法随机访问序列中间的结点。

【考点延伸】排序算法

4 【正解】B

【解析】

【考点延伸】二叉树的非递归遍历

5 【正解】C

【解析】希尔排序是根据增量将序列划分成一个个不连续的单位，在每个单位中使用插入排序，直到增量变为 1。第一趟结果是 49,13,27,50,76,38,65,97

【考点延伸】希尔排序

6 【正解】B

【解析】四叉树中每一个结点最多有 4 个孩子。因此边数+1 = 度为 4 的结点数+度为 3 的结点数+度为 2 的结点数+度为 1 的结点数+叶子结点数； 而度为 1 的结点数+2*度为 2 的结点数+3*度为 3 的结点数+4*度为 4 的结点数 = 边数。叶子结点数为 20

7 【正解】A

【解析】当左右指针指向的元素和 key 相等时立即停下并交换，会交换 N/2 次,然后递归左右子区间,需要递归 logN 次，故时间复杂度为 O(NlogN)

【考点延伸】快速排序，时间复杂度

8 【正解】A

【解析】2435-2047=388，1024+388/2 = 1218

【考点延伸】完全二叉树

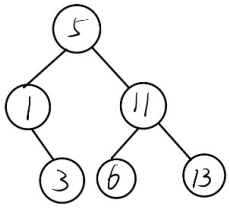
9 【正解】D

【解析】通过迭代可以计算出 P1: $T(n) = log_3 n + 1$,P2: $T(n) = \frac{3}{2}n - \frac{1}{2}$ 因此 $O(\log N)$ for P1, $O(N)$ for P2

【考点延伸】时间复杂度

10【正解】A

【解析】二叉搜索树如下图所示



【考点延伸】二叉搜索树

11【正解】D

【解析】第一遍先按个位排序，→431→331→33→63→24→56→46→57→7→28

【考点延伸】基数排序

12 【正解】C

【解析】已知循环队列的大小，在数组中能存储的最大元素数量就是数组的大小 m

【考点延伸】循环队列

13【正解】D

【解析】对于链表的操作，画图是最容易理解的。在 p 后插入 s，先把 p 后面的结点 q 的前驱结点指向 s，再把 p 的下一个结点指向 s, 然后把 s 的前驱结点指向 p，s 的后继结点指向 q

【考点延伸】双向循环链表

三、程序填空题（满分 15 分）

1、【解析】(1) j < n – i + 1

(2) r[j]->key < r[mini]->key

(3) maxi==i && mini !=i

【考点延伸】简单选择排序的改进

2、【解析】(1) H[child] < H[child + 1] (5 分)

(2) H[child] > H[0](5 分)

【考点延伸】堆排序

四、函数题(满分 6 分)

【解析】

```
int a[100];
int i=0;
void pretravel(Tree T){
    if(!T)
        return;
    if (T->Left)
        pretravel(T->Left);
```

```
        a[i]=T->Element;
        i++;
        if (T->Right)
            pretravel(T->Right);
        return;
}
void Print_NGT( Tree T,    int X ){
        pretravel(T);
        for (int j=i-1; j>=0; j--) {
            if(a[j]<=X)
            printf("%d ",a[j]);
        }
}
```

【考点延伸】二叉搜索树