

期末复习练习题3

开始时间2023/12/18 15:02:00

结束时间2024/01/13 17:02:00

答题时长37560分钟

答卷类型标准答案

总分50

填空题

得分：暂无 总分：24

4-1 设有一组关键字{9,1,23,14,55,20,84,27}，采用哈希函数：H (key) =key mod 7，表长为10，用开放地址法的线性探测再散列方法解决冲突。要求：对该关键字序列构造哈希表。计算查找成功,失败的平均查找长度。依次给出哈希表地址0--9单元的值。

14

(1分)

1

(1分)

9

(1分)

23

(1分)

84

(1分)

|

(1分)

55

(1分)

20

(1分)

27

(1分)

|

(1分)

平均查找长度：ASLsucc=

16/8|2

(2分)

ASLunsucc=

31/10|3.1

(2分)

4-2 已知哈希表长度为 8，哈希函数为 $H(k) = k \bmod 7$ ，采用线性探测法处理冲突。将下列关键字依次插入到哈希表中，

34, 37, 20, 16, 13

请写出存入数据后的哈希表。

0	13	(1分)
1	-	(1分)
2	37	(1分)
3	16	(1分)
4	-	(1分)
5	-	(1分)
6	34	(1分)
7	20	(1分)

注：空白处填“-”。

若各关键字的检索概率相等，则平均查找长度为

1.8

(2分)

(保留1位小数，末位四舍五入)。

程序填空题

得分：暂无 总分：11

5-1 二分查找。

```
#include<iostream>
using namespace std;
#define MAXSIZE 100

typedef struct{
    int key;
}ElemType;
```

```

typedef struct{
    ElemType *R;
    int length;
}SSTable;

int Create_SSTable(SSTable &L)
{
    int n;
    cin >> n;
    for(int i=1;i<=n;i++)
    {
        cin >> L.R[i].key;
        L.length++;
    }
    return 1;
}

int Search_Bin(SSTable ST,int key) {
    int low=1,high=ST.length;
    int mid;
    while(low<=high (2分)) {
        mid=(low+high) / 2;
        if (key==ST.R[mid].key (2分)) return mid;
        else if (key<ST.R[mid].key (2分)) high = mid -1;
        else low =mid +1;
    }
    return 0;
}

int main()
{
    SSTable ST;
    int key;
    int result;
    ST.R=new ElemType[MAXSIZE];
    ST.length=0;
    Create_SSTable(ST);
    cin >> key;
    result=Search_Bin(ST, key);
    if(result)
        cout << "search success, The key is located in "<< result;
    else
        cout << "search failed";
    return 0;
}

```

输入样例:

第一行输入一个数n，第二行输入n个数，第三行输入要查找的数key。

```

11
5 13 19 21 37 56 64 75 80 88 92
21

```

输出样例:

```
search success, The key is located in 4
```

5-2 顺序查找。

```
#include<iostream>
using namespace std;
#define MAXSIZE 100
#define OK 1;

typedef struct{
    int key;
}ElemType;

typedef struct{
    ElemType *R;
    int length;
}SSTable;

int Create_SSTable(SSTable &L)
{
    int n;
    cin >> n;
    for(int i=1;i<=n;i++)
    {
        cin >> L.R[i].key;
        L.length++;
    }
    return 1;
}

int Search_Seq(SSTable ST, int key){
    int i;
    ST.R[0].key (2分) = key;
    for(i = ST.length; ST.R[i].key!=key (3分); --i);
    return i;
}

int main()
{
    SSTable ST;
    int key;
    int result;
    ST.R=new ElemType[MAXSIZE];
    ST.length=0;
    Create_SSTable(ST);
    cin >> key;
    result=Search_Seq(ST, key);
    if(result)
        cout << "search success";
    else
        cout << "search failed";
    return 0;
}
```

```
}
```

输入样例:

第一行输入一个数n，第二行输入n个数，第三行输入要查找的数key。

```
7
24 53 45 45 12 24 90
24
```

输出样例:

若查找成功，输出“search success”，查找失败，输出“search failed”。

```
search success
```

编程题

得分：暂无 总分：15

7-1 快速排序的过程 (15分)

给定n个整型元素，利用快速排序算法对其进行非递减排序，请输出每一趟Partition的结果。每次选择所处理的区间的第一个元素作为基准元素。

输入格式:

输入为两行，第一行为一个整数n ($1 < n \leq 1000$)，表示元素个数。第二行为n个空格间隔的整数，表示待排序的元素。

输出格式:

输出为若干行，每行依次输出Partition后的结果，每个元素后一个空格。

输入样例:

```
5
4 5 3 2 1
```

输出样例:

```
2 1 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```