



本小节内容

2022 年 42 题代码实战

18 节课课程内容评价

2022 年 42 题代码实战

42. (10 分)

现有 n ($n > 100000$) 个数保存在一维数组 M 中, 需要查找 M 中最小的 10 个数, 请回答下列问题。

(1) 设计一个完成上述查找任务的算法, 要求平均情况下的比较次数尽可能少, 简单描述其算法思想, 不需要程序实现。

(2) 说明你所设计的算法平均情况下的时间复杂度和空间复杂度。

答案解析:

这道题非常有代表性, 考研并没有要求代码实现, 我们这里给大家实战一下堆排的代码, 从而使大家理解的更加深刻

方法二: 堆 (堆排序思想)

定义含 10 个元素的大根堆 H , 元素值均为该堆元素类型能表示的最大数 MAX 。

for M 中的每个元素 s

if ($s < H$ 的堆顶元素) 删除堆顶元素并将 s 插入到 H 中;

当数据全部扫描完毕, 堆 H 中保存的即是最小的 10 个数。

2) 算法平均情况下的时间复杂度是 $O(n)$, 空间复杂度是 $O(1)$ 。

代码如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
typedef int ElemType;
typedef struct{
    ElemType *elem;
    int TableLen;
}SSTable;

void ST_Init(SSTable &ST,int len)//申请空间, 并进行随机数生成
{
    ST.TableLen=len;
```



```
ST.elem=(ElemType *)malloc(sizeof(ElemType)*ST.TableLen);
int i;
srand(time(NULL));
for(i=0;i<ST.TableLen;i++)
{
    ST.elem[i]=rand()%100;
}
}
void ST_print(SSTable ST)
{
    for(int i=0;i<10;i++)
    {
        printf("%3d",ST.elem[i]);
    }
    printf("\n");
}
void swap(ElemType &a,ElemType &b)
{
    ElemType tmp;
    tmp=a;
    a=b;
    b=tmp;
}

//调整子树
void AdjustDown(ElemType A[], int k, int len)
{
    int dad = k;
    int son = 2 * dad + 1; //左孩子下标
    while (son<=len)
    {
        if (son + 1 <= len && A[son] < A[son + 1])//看下有没有右孩子，比较左右孩子选大的
        {
            son++;
        }
        if (A[son] > A[dad])//比较孩子和父亲，如果孩子大于父亲，那么进行交换
        {
            swap(A[son], A[dad]);
            dad = son;//孩子重新作为父亲，判断下一颗子树是否符合大根堆
            son = 2 * dad + 1;
        }
        else {
            break;
        }
    }
}
```

```
    }  
}  
void HeapSort(ElemType A[], int len)  
{  
    int i;  
    //先对前 10 个元素建立大根堆  
    for (i = len/2; i >= 0; i--)  
    {  
        AdjustDown(A, i, len);  
    }  
    //比较剩余的 A[10]到 A[99999]元素，小于堆顶，就放入 A[0],继续调整 10 个元素为大根堆  
    for (i = 10; i < 100000; i++)  
    {  
        if (A[i] < A[0])  
        {  
            A[0] = A[i];  
            AdjustDown(A, 0, 9); //继续调整为大根堆  
        }  
    }  
}  
  
//《王道 C 督学营》课程  
//2022 年 42 题真题实战  
int main()  
{  
    SSTable ST;  
    ST_Init(ST, 100000); //初始化  
    HeapSort(ST.elem, 9);  
    ST_print(ST);  
    return 0;  
}
```

18 节课课程内容评价

