基础作业

1 什么是算法的时间复杂度？什么是算法的空间复杂度

- 时间复杂度
  - 算法运行所需要的时间
- 空间复杂度
  - 算法运行所占用临时空间的大小

2 八大排序是哪八大排序？

- \选择\插入\希尔\快排\堆\归并\基数

3 实现冒泡排序

```
#define N 1024;
void bubbleSort(int *Arr)
{
    for(int i=0;i<=N;i++)
    {
        for(int j=0;j<N-i;j++)
        {
            if(Arr[i]>Arr[i+1])
            {
                swap(Arr[i],Arr[i+1]);
            }
        }
    }
}
```

4 证明冒泡排序的正确性

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstdlib>
#include<ctime>
void bubbleSort(int* Arr);
void arrPrint(int* arr);
using namespace std;
int main()
{
    int Arr[10];
    srand(time_t(NULL)); //随机数种子
    time_t start, end;
    for (int i = 0; i < 10; i++)
    {
        Arr[i] = rand() % 100;
    }
    cout << "原数据为" << endl;
    arrPrint(Arr);
```

```
    cout << "-----------------------------" << endl;
    bubbleSort(Arr);
    cout << "排序后为" << endl;
    cout << "-----------------------------" << endl;
    arrPrint(Arr);
}

void bubbleSort(int* a)
{
    for (int i = 0; i <= 10-1; i++)
    {
        for (int j = 0; j < 10 -1-i; j++)
        {
            if (a[j] > a[j + 1])
            {
                swap(a[j], a[j + 1]);
            }
        }
    }
}
void arrPrint(int* arr)
{
    for (int i = 0; i < 10-1; i++)
    {
        printf("%3d", arr[i]);
    }
    printf("\n");
}
```

5 画图说明插入排序、选择排序和希尔排序的过程

- 插入排序
- 选择排序
- 希尔排序

6 重排一个数组，以a[n-1]为标准，所有小于标准的数字放在标准左边，所有大于标准的数字放在标准的右边。

```
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstdlib>
#include<ctime>
void bubbleSort(int* Arr);
void arrPrint(int* arr);
using namespace std;
int main()
{
```

```
    int Arr[10];
    srand(time_t(NULL)); //随机数种子
    time_t start, end;
    for (int i = 0; i < 10; i++)
    {
        Arr[i] = rand() % 100;
    }
    int pivotVal=Arr[9];
    cout << "原数据为" << endl;
    arrPrint(Arr);
    cout << "------------------------------" << endl;
    partition(Arr,pivotVal);
    cout << "排序后为" << endl;
    cout << "------------------------------" << endl;
    arrPrint(Arr);
}
void partition (int *N,int pivotVal) //设这里数组长度为10
{
    int begin=0; //头指针
    int end=9;  //尾指针
    int pivot=pivotVal;
    while(being<end)
    {
        if(N[begin]<=pivot)
        {//如果后面发现大于pivot的值则向前移动指针
            begin++;
        }
        else if//如果比pivot大，则直接交换此时end和bigin的值
        {
            N[begin]=N[end];
        }
        else if(N[end]>=pivot)
        {//如果后面发现大于pivot的值则向前移动指针
            --end;
        }
         else//如果比pivot大，则直接交换此时end和bigin的值
        {
            N[end]=N[begin];
        }
    }
}
```

-

7 说明冒泡、选择和插入排序的最坏和平均时间复杂度

排序方法	最坏时间复杂度	平均时间复杂度
冒泡	O(n^2)	O(n^2)
选择	O(n^2)	O(n^2)
插入	O(n^2)	O(n^2)

8 说明快速排序最坏和平均时间复杂度

- 快速排序时间复杂度最坏为O(n^2)
- 快速排序时间复杂度平均为O(nlogn)