《数据结构与算法》实验报告

实验名称	算法复杂度比较				
姓名	叶鹏	学号	20020007095	日期	2022/2/25
实验内容	动手撰写随机数生成算法，生成一组随机数 通过主函数调用两个排序算法，对随机数进行排序，并打印结果和运行时间				
实验目的	根据运行时间的不同，体会算法复杂度的影响				

实验步骤

1. 定义数组(假定 10k 数据量)，获取用户输入的总量 n，生成 n 位随机数(0 <= num < 1000)

```
42  int main()
43  {
44      int *arr = new int[100000];
45
46      int n;
47      cout << "enter the amount:" << endl;
48      cin >> n;
49
50      //get random numbers
51      srand(unsigned(time));
52
53      for (int i = 0; i < n; i++)
54      {
55          arr[i] = rand() % 1000;
56      }
57
```

2. 导入 ppt 提供的两种排序算法

```
6   void BubbleSort(int a[], int n)
7   {
8       int i, j, temp;
9       for (i = 0; i < n - 1; i++)
10      {
11          for (j = 0; j < n - 1 - i; j++)
12          {
13              if (a[j] > a[j + 1])
14              {
15                  temp = a[j];
16                  a[j] = a[j + 1];
17                  a[j + 1] = temp;
18              }
19          }
20      }
21  }
```

```
23  void QuickSort(int a[], int left, int right)
24  {
25      if (left >= right)
26          return;
27      int i = left, j = right, key = a[left];
28      while (i < j)
29      {
30          while (i < j && key <= a[j])
31              j--;
32          a[i] = a[j];
33          while (i < j && key >= a[i])
34              i++;
35          a[j] = a[i];
36      }
37      a[i] = key;
38      QuickSort(a, left, i - 1);
39      QuickSort(a, i + 1, right);
40  }
```

3. 设定计时器记录两种算法的运行时间，分别用两种算法对数据进行排序，观察执行时间

```
    //set the clock
    clock_t start, end;
    start = clock();

    //bubble sort
    BubbleSort(arr, n);

    end = clock();

    cout << "running time: " << (double)(end - start) / CLOCKS_PER_SEC << "s" << endl;

    //quick sort
    start = clock();
    QuickSort(arr, 0, n - 1);

    end = clock();

    cout << "running time: " << (double)(end - start) / CLOCKS_PER_SEC << "s" << endl;

    return 0;
```

4. 测试数据

1) 20000

```
enter the amount:
20000
running time: 0.756s [BubbleSort]
running time: 0.002s [QuickSort]
```

2) 50000

```
enter the amount:
50000
running time: 5.243s [BubbleSort]
running time: 0.004s [QuickSort]
```

3) 90000

```
enter the amount:
90000
running time: 16.977s [BubbleSort]
running time: 0.014s [QuickSort]
```

5. 原始代码

```cpp
#include <iostream>
#include <cstdlib>
#include <time.h>
using namespace std;

void BubbleSort(int a[], int n)
{
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - 1 - i; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}

void QuickSort(int a[], int left, int right)
{
    if (left >= right)
        return;
    int i = left, j = right, key = a[left];
    while (i < j)
    {
        while (i < j && key <= a[j])
            j--;
        a[i] = a[j];
        while (i < j && key >= a[i])
            i++;
        a[j] = a[i];
    }
    a[i] = key;
    QuickSort(a, left, i - 1);
    QuickSort(a, i + 1, right);
}

int main()
{
    int *arr = new int[100000];
    int *arr_cp = new int[100000];

    int n;
    cout << "enter the amount:" << endl;
    cin >> n;

    //get random numbers
    srand(unsigned(time));

    for (int i = 0; i < n; i++)
    {
        arr[i] = rand() % 1000;
        arr_cp[i] = arr[i];
    }

    //set the clock
    clock_t start, end;
    start = clock();

    //bubble sort
    BubbleSort(arr, n);

    end = clock();

    cout << "running time: " << (double)(end - start) / CLOCKS_PER_SEC << "s [BubbleSort]" << endl;

    // cout << "result:" << endl;
    // for (int i = 0; i < n; i++)
    // {
    //     cout << arr[i] << " ";
    // }
    // cout << endl;

    //quick sort
    start = clock();
    QuickSort(arr_cp, 0, n - 1);

    end = clock();

    cout << "running time: " << (double)(end - start) / CLOCKS_PER_SEC << "s [QuickSort]" << endl;

    // cout << "result:" << endl;
    // for (int i = 0; i < n; i++)
    // {
    //     cout << arr_cp[i] << " ";
    // }
    // cout << endl;

    delete arr;
    delete arr_cp;

    return 0;
}
```

实验总结	本次实验通过对两种排序算法进行实际观察，体会到数据规模对算法执行时间的影响，以及不同的算法之间的时间复杂度的差异。