第一题、排序算法

1. #include <iostream>
2. #include <cstdlib>
3. using namespace std;
4. *//冒泡排序,时间复杂度n\*n=O(n^2),空间复杂度O(n)*
5. void BubbleSort(int\* arr, int n) {
6. int i, j;
7. for (i = 0; i < n-1; i++) {*//整体遍历*
8. for (j = 0; j < n-i-1; j++) {*//内层遍历*
9. if (arr[j] > arr[j+1]) {*//前后大小交换*
10. int temp = arr[j];
11. arr[j] = arr[j+1];
12. arr[j+1] = temp;
13. }
14. }
15. }
16. }
17. *//选择排序,时间复杂度n\*n=O(n^2),空间复杂度O(n)*
18. void SelectionSort(int\* arr, int n) {
19. int i, j, min\_idx;
20. *//一共要进行n-1次选择排序*
21. for (i = 0; i < n - 1; i++) {
22. *// 查找最小值的索引*
23. min\_idx = i;
24. for (j = i + 1; j < n; j++) {
25. if (arr[j] < arr[min\_idx]) {
26. min\_idx = j;
27. }
28. }
29. *// 将找到的最小值交换到序列头部*
30. int temp = arr[min\_idx];
31. arr[min\_idx] = arr[i];
32. arr[i]= temp;
33. }
34. }
35. *//插入排序，时间复杂度n\*n=O(n^2),空间复杂度O(n)*
36. void InsertionSort(int \*arr,int n)
37. {
38. int temp;
39. for (int i = 0; i < n-1; ++i) {
40. for (int j = i+1; j > 0; --j) {
41. if(arr[j]<arr[j-1])
42. {
43. temp = arr[j];
44. arr[j]=arr[j-1];
45. arr[j-1]= temp;
46. }
47. }
48. }
49. }
50. *//根据用户需求动态分配数组*
51. int\* ArrayAllocate(int n)
52. {
53. int \*arr = (int\*)malloc(n \* sizeof(int));*//分配内存*
54. if(arr == nullptr)*//检查是否分配成功*
55. {
56. cout << "内存分配失败" << endl;
57. exit(0);
58. }
59. cout<<"请输入" << n <<"个数字:" << endl;*//输入数字储存到数组*
60. for (int i = 0; i < n; ++i) {
61. cin >> arr[i];
62. }
63. cout << "你输入的数组为" << endl;
64. for (int i = 0; i < n; ++i) {
65. cout<< arr[i] <<" ";
66. }
67. cout <<endl;
68. return arr;
69. }
70. *//打印数组*
71. void PrintArray(int \*arr,int n)
72. {
73. cout << "排序后的数组为" << endl;
74. for (int i = 0; i < n; ++i) {
75. cout << arr[i] << " ";
76. }
77. }
78. int main()
79. {
80. int length =0;
81. cout <<  "请输入你要输入数据的数量:"<<endl;*//用户输入数字的数量*
82. cin >> length;
83. int\* arr;
84. arr=ArrayAllocate(length);
85. cout << "请选择你需要的算法" <<endl<<"1.冒泡排序"<<endl<<"2.选择排序"<<endl<<"3.插入排序"<<endl;
86. int choice =0;
87. Switch:
88. cin >> choice;
89. switch (choice) {
90. case 1 :
91. BubbleSort(arr,length);
92. break;
93. case 2 :
94. SelectionSort(arr,length);
95. break;
96. case 3 :
97. InsertionSort(arr,length);
98. break;
99. default:
100. cout << "请输入正确的数字" <<endl;
101. goto Switch;
102. }
103. PrintArray(arr,length);
104. return 0;
105. }

输入输出样例

请输入你要输入数据的数量:

5

请输入5个数字:

23

47

10

3

56

你输入的数组为

23 47 10 3 56

请选择你需要的算法

1.冒泡排序

2.选择排序

3.插入排序

1

排序后的数组为

3 10 23 47 56

第二题、汉诺塔问题//时间复杂度O（2^n）,空间复杂度O（n！）

1. #include <iostream>
2. using namespace std;
3. void Hanoi(int n,char from, char tmp, char to)
4. {
5. if(n == 1){
6. cout << "把盘1从"<<from<<"移到"<< to <<endl;
7. return;
8. }
9. Hanoi(n-1,from, to,tmp);
10. cout<<"把盘"<<n<<"从"<<from<<"移到"<<to<<endl;
11. Hanoi(n-1,tmp,from,to);
12. }
13. int main() {
14. int n;
15. cout <<"请输入盘子数量"<<endl;
16. cin >> n;
17. Hanoi(n,'A','B','C');
18. return 0;
19. }

输入输出样例

请输入盘子数量

5

把盘1从A移到C

把盘2从A移到B

把盘1从C移到B

把盘3从A移到C

把盘1从B移到A

把盘2从B移到C

把盘1从A移到C

把盘4从A移到B

把盘1从C移到B

把盘2从C移到A

把盘1从B移到A

把盘3从C移到B

把盘1从A移到C

把盘2从A移到B

把盘1从C移到B

把盘5从A移到C

把盘1从B移到A

把盘2从B移到C

把盘1从A移到C

把盘3从B移到A

把盘1从C移到B

把盘2从C移到A

把盘1从B移到A

把盘4从B移到C

把盘1从A移到C

把盘2从A移到B

把盘1从C移到B

把盘3从A移到C

把盘1从B移到A

把盘2从B移到C

把盘1从A移到C

第三题、角谷猜想

1. #include <iostream>
2. using namespace std;
3. int main() {
4. int maxround = 0;
5. int maxcount = 0;
6. int n = 0;
7. for (int i = 1; i <= 100 ; ++i) {
8. int count =0;
9. n=i;
10. do {
11. if(n%2==0)
12. {
13. n=n/2;
14. }
15. else n=n\*3+1;
16. count ++;
17. }while(n!=1);
18. if(count > maxcount){maxcount = count;
19. maxround = i;}
20. }
21. do {
22. cout << maxround <<" ";
23. if(maxround%2==0)
24. {
25. maxround=maxround/2;
26. }
27. else maxround=maxround\*3+1;
28. }while(maxround!=1);
29. cout <<  "1";
30. return 0;
31. }

输入输出样例

97 292 146 73 220 110 55 166 83 250 125 376 188 94 47 142 71 214 107 322 161 484 242 121 364 182 91 274 137 412 206 103 310 155 466 233 700 350 175 526 263 790 395 1186 593 1780 890 445 1336 668 334 167 502 251 754 377 1132 566 283 850 425 1276 638 319 958 479 1438 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367 4102 2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 488 244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1