Report For Find The Minimum

09019204
曹邹颖

1. Problem description / demand analysis 问题描述

Given a list of numbers, we know that the numbers are decreasing, then increasing. The goal is to find the smallest number in the list. Of course, you go through all the numbers of the list. Design a faster algorithm to find the smallest number in the list.

1) Suppose that any two consecutive numbers are different in the list. How many comparisons do you need to find the minimum if n=20? If n=100?
L = [20,18,14,13,12,9,10,12,14,15,16,20,25,30]
min(L) = 9

2) Suppose that two consecutive numbers may be the same in the list. How do you handle this case? How many comparisons do you need to find the minimum if n=20? If n=100?
L = [20,18,14,13,12,9,10,12,14,15,15,15,18,20,25,30]

2. System structure / algorithm idea 算法设计

(1) basic idea 基本思想

除了最基本的遍历方法，首先我构思的便是二分法，鉴于序列先非严格递减后非严格递增的特征，二分时中间两个数比较，便可以得到最小值位于哪一半部分，这样采用递归算法便可以求解，算法复杂度在 O(logn)。

(2) system framework 代码框架

构造一个递归函数 int findTheMinimum(int* num, int left, int right);传入一个 num 数组，比较位置为下标 left 到 right,返回其中最小值，若中间两个数左边数大，则在右半部分查找最小值，否则在左半部分查找最小值，递归截止条件为 left==right 即为一个数, 直接返回该数递归结束，对于相等情况处理细节见 Report 小节 3.。

(3) the functions and relationships of each module 模块的功能和关系

该算法一个递归函数即可，无多个模块。

3. Recursive function design 递归函数设计

(1) recursive function design idea

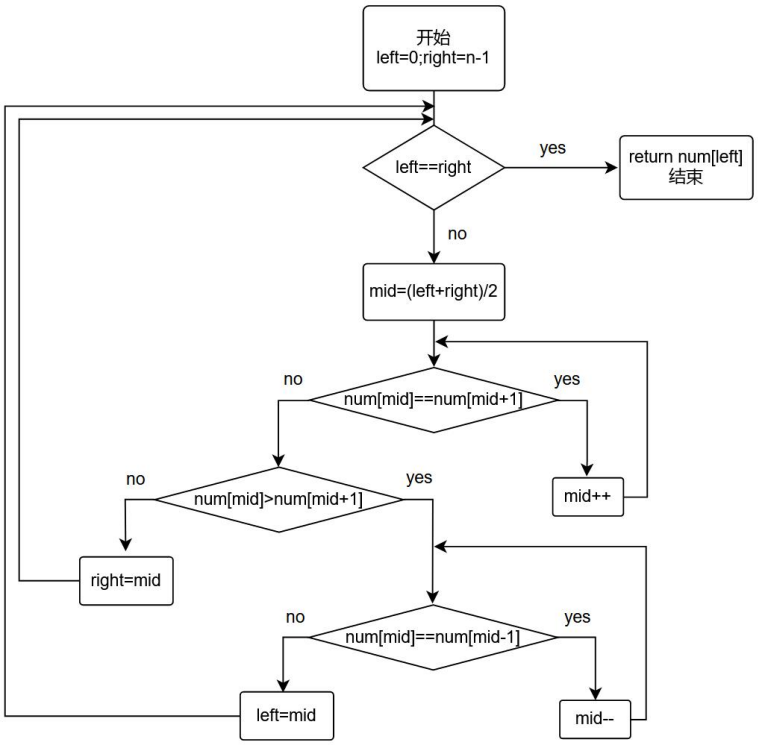
一开始未考虑存在相等情况：
int findTheMinimum(num, left, right)

若 left==right
　　返回 num[left]
不然，取中间数 mid = (right + left) / 2
　　如果 num[mid] < num[mid + 1]
　　　　调用 findTheMinimum(num, left, mid)在左半部分找最小值
　　否则
　　　　调用 findTheMinimum(num, mid + 1, right)在右半部分找最小值

考虑存在相等情况：
int findTheMinimum_1(num, left, right)
　　若 left==right
　　　　返回 num[left]
　　不然，取中间数 mid = (right + left) / 2
　　通过和右边的数比较找到相等数的最右边一个
　　　　采用 while (num[mid] == num[mid + 1])mid++;循环
　　此时如果 num[mid] > num[mid + 1]
　　　　则需回到相等数的最左边一个
　　　　　　采用 while (num[mid] == num[mid - 1])mid−−;循环
　　　　调用 findTheMinimum(num, mid, right)在左半部分找最小值
　　否则
　　　　调用 findTheMinimum(num, left, mid)在右半部分找最小值

(2) flow chart

考虑存在相等情况：



(3) algorithm complexity analysis

不存在相等情况下：

算法复杂度 T(n)=T(n/2)+1→O(logn)。

存在相等情况下：

最差情况是均为相等的数则 O(n/2)=O(n)，一般情况下算法 T(n)=T(n/2)+1→O(logn)。

4. Test results and analysis 测试结果和分析

(1) test data selection or generation method 测试数据选择/生成方法

首先测试：

本题第一问：n=14, L = [20,18,14,13,12,9,10,12,14,15,16,20,25,30]

本题第二问：n=16, L = [20,18,14,13,12,9,10,12,14,15,15,15,18,20,25,30]

接着测试：

是否考虑存在相等情况的两种情况下，n=20，n=100 数据

生成方法：

不存在相等情况下：

使用随机生成 n 个不重复的且排序为先非严格递减后非严格递增的值，以最小值排在第 1 到 n 个位置生成 n 组测试数据。

testData.cpp | findTheMinimum.cpp | investmentProblem.cpp

GenerateTestData | (全局范围)

```cpp
#include<iostream>
#include<algorithm>
#include<vector>
#include<stdio.h>
#include<time.h>
#include<fstream>
using namespace std;
int main()
{
    int n,index;
    cin >> n;
    index = n;
    vector<int> t;
    srand(time(0));
    ofstream file;
    file.open("testData_20.txt");
    while (index) {
        int bias = rand() % 100 + 1;
        for (int i = index - 1; i >= 0; i--)
            file << i + bias << " ";
        for (int i = index; i < n; i++)
            file << i + bias << " ";
        file << '\n';
        index--;
    }
    return 0;
}
```

存在相等情况下：

使用随机生成 n 个重复的且排序为先非严格递减后非严格递增的值，由于使用 rand()范围较小会生成重复随机数，从而生成 n 组测试数据。

testData.cpp | findTheMinimum.cpp | investmentProblem.cpp

GenerateTestData | (全局范围)

```cpp
#include<iostream>
#include<algorithm>
#include<vector>
#include<stdio.h>
#include<time.h>
#include<fstream>
using namespace std;
int main()
{
    int n,index; cin >> n; index = n;
    vector<int> t;
    srand(time(0));
    ofstream file; file.open("testData_100_r.txt");
    while (index) {
        for (int i = 0; i < index; i++)
            t.push_back(rand() % (n/2) + 1);
        sort(t.begin(), t.end());
        for (int i = index - 1; i >= 0; i--)
            file << t[i] << " ";
        int min = t.front();
        t.clear();
        for (int i = index; i < n; i++)
            t.push_back(min + rand() % (n/2) + 1);
        sort(t.begin(), t.end());
        for (int i = 0; i < n - index; i++)
            file << t[i] << " ";
        file << '\n';
        index--;
    }
    return 0;
}
```

(2) operation result screenshot 运行结果截屏

不存在相等情况下：

本题第一问：n=14,L = [20,18,14,13,12,9,10,12,14,15,16,20,25,30]

Microsoft Visual Studio 调试控制台

```
Input the number of numbers: n = 14
20 18 14 13 12 9 10 12 14 15 16 20 25 30
The minimum number is: 9
The number of comparisons I need to find the minimum if n = 14 is: 4
```

n=20：

通过 4.(1)生成方法生成的各样测试数据得到每次的比较次数，并算出平均值

Microsoft Visual Studio 调试控制台

```
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 50
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 77
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 34
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 79
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 92
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 80
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 10
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 98
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 62
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 4
The minimum number is: 59
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 25
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5

The average number of comparisons the algorithm needs to find the minimum if n = 20 is: 4.4
```

可见，n=20 时比较次数为 4/5，平均下来 4.4；

n=100：

通过 4.(1)生成方法生成的各样测试数据得到每次的比较次数，并算出平均值

Microsoft Visual Studio 调试控制台

```
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 81
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 22
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 6
The minimum number is: 87
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 30
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 71
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 6
The minimum number is: 48
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 2
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 73
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 81
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 77
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 34
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7

The average number of comparisons the algorithm needs to find the minimum if n = 100 is: 6.72
```

可见，n=100 时比较次数为 6/7，平均下来 6.72；

存在相等情况下：

本题第二问：n=16,L = [20,18,14,13,12,9,10,12,14,15,15,15,18,20,25,30]

Microsoft Visual Studio 调试控制台

```
Input the number of numbers: n = 16
20 18 14 13 12 9 10 12 14 15 15 15 18 20 25 30
The minimum number is: 9
The number of comparisons the algorithm needs to find the minimum if n = 16 is: 5
```

n=20：

通过 4.(1)生成方法生成的各样测试数据得到每次的比较次数，并算出平均值

Microsoft Visual Studio 调试控制台

```
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 8
The minimum number is: 1
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 6
The minimum number is: 2
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 1
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 6
The minimum number is: 1
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 2
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 8
The minimum number is: 2
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 6
The minimum number is: 6
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 8
The minimum number is: 4
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5
The minimum number is: 3
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 6
The minimum number is: 2
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 7
The minimum number is: 7
The number of comparisons the algorithm needs to find the minimum if n = 20 is: 5

The average number of comparisons the algorithm needs to find the minimum if n = 20 is: 7.2
```

可见，n=20 时比较次数平均下来 7.2；

n=100：

通过 4.(1)生成方法生成的各样测试数据得到每次的比较次数，并算出平均值

Microsoft Visual Studio 调试控制台

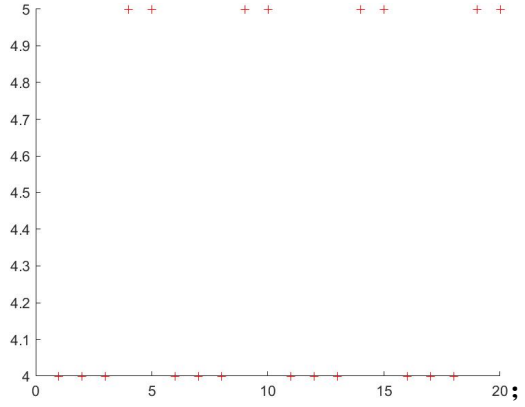
```
The minimum number is: 5
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 8
The minimum number is: 8
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 19
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 8
The minimum number is: 22
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 10
The minimum number is: 11
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 9
The minimum number is: 24
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 10
The minimum number is: 25
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 7
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 11
The minimum number is: 10
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 29
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 34
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7
The minimum number is: 31
The number of comparisons the algorithm needs to find the minimum if n = 100 is: 7

The average number of comparisons the algorithm needs to find the minimum if n = 100 is: 9.86
```

可见，n=100 时比较次数平均下来 9.86；

(3) performance diagram 性能图

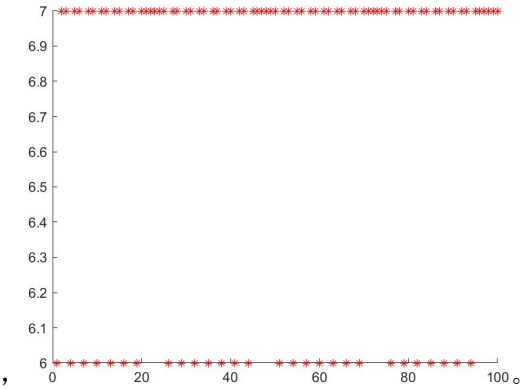
不存在相等情况下，测试数据组的比较次数：算法复杂度 O(logn)

n=20 时，



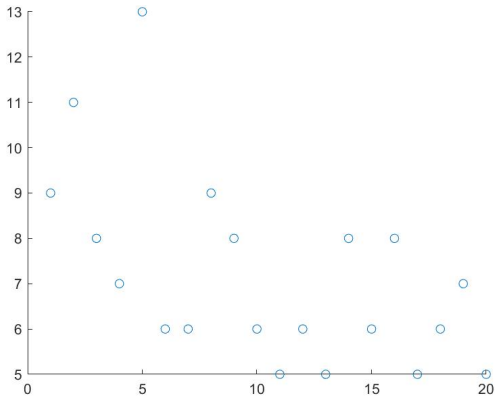
；

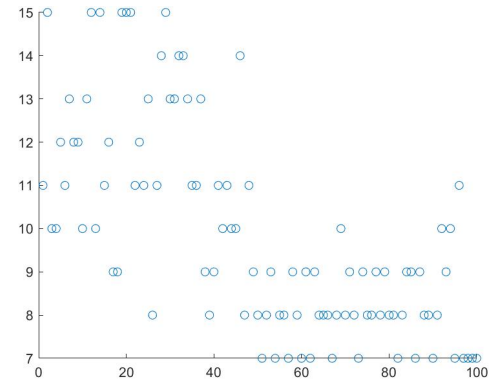
n=100 时，



。

存在相等情况下，测试数据组的比较次数：

n=20 时， 

n=100 时， 

5. Experimental summary 实验总结

(1) the problems encountered 遇到的问题

问题：

A. 递归算法出现死循环的情况;

B. 对于非严格递减与非严格递增存在相等情况的处理问题;

C. 对于计算算法复杂度随机生成测试数据的方法应使得最终测试结果具有普遍性;

(2) the problem-solving process 解决方案

解决方案：

A. 加上递归截止条件为 left+1==right 即两个相邻的数时;

B. 通过遍历找到相等的几个数的最左边/右边一个，再进行递归，详见 Report 小节 3.(1);

C. 考虑影响算法复杂度的参数，多次变化参数生成测试数据，详见 Report 小节 4.(1)。

(3) summarize the experimental experience 实验总结

实验中，理解了分治算法的概念和基本要素，学会分析二分算法的时间和空间复杂度，二分法能够实现 O(logn)的时间复杂度，比一般的顺序搜索快了不少，同时理解了递归的概念，明确了添加递归结束条件这一步骤的重要性。

在不存在相等情况下时，我很容易想到采用二分法，易知其算法复杂度为O(logn)，应该算是比较快的算法了，而考虑存在相等情况下时，我首先做出的算法修改是在二分法的基础上遍历那些相等的数找到边界数。

6. Source code 源代码

```
#include<iostream>
#include<math.h>
#include<fstream>
using namespace std;
int findTheMinimum(int* num, int l, int r);        // 不考虑存在相等情况
int findTheMinimum_1(int* num, int l, int r);      // 考虑存在相等情况
int m;                                             // m:the number of comparisons in the algorithm
int main()
{
    int n;                                         // n:the number of numbers
    cout << "Input the number of numbers: n = ";
    cin >> n;
    int* num = new int[n];
    /*for (int i = 0; i < n; i++)
        cin >> num[i];
    cout << "The minimum number is: ";
    cout << findTheMinimum(num, 0, n - 1) << endl;
    cout << "The number of comparisons the algorithm needs to find the minimum if n = " << n << "
        is: " << m << endl;*/
    fstream infile("testData_100_r.txt", ios::in);
    if (!infile)
    {
        cout << "open error!" << endl;
        exit(1);
    }
    int index = n,total=0;
    while (index) {                                // index 组测试数据
        m = 0;
        for (int i = 0; i < n; i++)
            infile >> num[i];
        cout << "The minimum number is: ";
        cout << findTheMinimum_1(num, 0, n - 1) << endl;
        cout << "The number of comparisons the algorithm needs to find the minimum if n = " << n
<< " is: " << m << endl;
        index --;
        total += m;
    }
    cout<< "\nThe average number of comparisons the algorithm needs to find the minimum if n = "
<< n << " is: " << double(total)/n << endl;
```

```
    delete[]num;
    return 0;
}
int findTheMinimum(int* num, int l, int r)              // 一开始未考虑存在相等情况
{
    if (l == r)
        return num[l];                                  // 递归截止
    int mid = (r + l) / 2;
    m++;// for if followed
    if (num[mid] < num[mid + 1])
        return findTheMinimum(num, l, mid);             // 找左半部分
    else
        return   findTheMinimum(num, mid + 1, r);// 找右半部分
}
int findTheMinimum_1(int* num, int l, int r)            // 考虑存在相等情况
{
    if (l == r)
        return num[l];
    int mid = (r + l) / 2;
    m++;                                                // for while followed
    while (num[mid] == num[mid + 1]){                   // 找到相等数的最右边一个
        mid++;
        m++;
    }
    m++;                                                // for if followed
    if (num[mid] > num[mid + 1]) {
        m++;                                            // for while followed
        while (num[mid] == num[mid - 1]){               // 回到相等数的最左边一个
            mid--;
            m++;
        }
        return findTheMinimum(num, mid, r);             // 找右半部分
    }
    else
        return findTheMinimum(num, l, mid);             // 找左半部分
}
```