第二次上机讲解

郭延文 李元琪 费炀 李恣炜

2019年12月18日

评分规则

- 第一题6分，第二题9分
- OJ上通过的题目都是满分
- 没通过的助教会批改。第一题，如果仅仅是大矩阵样例时间超限，比如使用了六重循环，但是代码是正确的，小矩阵可以测试通过，只扣1分。如果思路整体错了，无论大矩阵还是小矩阵，都无法通过，酌情只给1-2分

题1：求矩形最大面积

注意读题！

输入一个只包含0和1的二维矩阵A，返回A中只包含1的最大矩形面积。其中，矩形的长和宽只能与A的行和列方向平行，不考虑对角线方向的情况。

输入

两个整数m和n，表示矩阵的行数和列数。$0 < m, n \leq 100$

接下来为m行正整数，每行n个，表示矩阵A

输出

一个整数，表示最大矩形面积

样例输入

```
1,0,1,0,0,

1,0,1,1,1,

1,1,1,1,1,

1,0,0,1,0
```

样例输出

6（最大矩形由样例中的粗体 **1** 组成）

还有两个测试样例，矩阵比较大（不过未超过100），在这里放不下，两个样例分别是：大部分地方是0，中间有一块矩形是1；大部分地方是0，中间有一个矩形是0

算法图示：

1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0

1 0 1 0 0
1 0 1 2 3
1 2 3 4 5
1 0 0 1 0

每个数字向上向下找，找到比它大的，说明可以组成更宽的矩形。这就可以比较大小
3往下找到5，说明矩形大小可以变成2*3=6，更新max
但是5往上找到3，3<5，矩形大小在这里还是5，不更新max

变形一下矩阵，让每个数字都代表它左边连续的1的数量。

```cpp
#include<iostream>
#define myMax(a, b) ((a) > (b) ? (a) : (b))
using namespace std;

int matrix[110][110];

int main() {
    int m, n;
    cin >> m >> n;

    for(int i = 0; i < m; ++i)
        for(int j = 0; j < n; ++j)
            cin >> matrix[i][j];

    for(int i = 0; i < m; ++i)
        for(int j = 1; j < n; ++j)
            if(matrix[i][j] != 0)
                matrix[i][j] += matrix[i][j-1];

    int rt = 0;
    for(int j = n - 1; j >= 0; --j) {
        for(int i = 0; i < m; ++i) {
            if(matrix[i][j] == 0) continue;
            int up = i - 1, down = i + 1, t = matrix[i][j];
            while(up >= 0 && matrix[up][j] >= t) up--;
            while(down < m && matrix[down][j] >= t) down++;
            rt = myMax(rt, t*(down-up-1));
        }
    }
    cout << rt << endl;
}
```

变形一下矩阵，让每个数字都代表它左边连续的1的数量

每个数字向上向下找，找到比它大的，说明可以组成更宽的矩形。这就可以比较大小

典型问题 未考虑空心的情况

```cpp
#include<iostream>
using namespace std;
int main() {
    int a[101][101];
    int m, n;
    cin >> m >> n;
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            cin >> a[i][j];

    int max = 0, max1 = 0, max2 = 0, max3 = 0;
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++) {
            for (int lin = i, col = j; a[lin][col] == 1 && col < n; col++) {
                max1++;
            }
            for (int lin = i, col = j; a[lin][col] == 1 && lin < m; lin++) {
                max2++;
            }
            max3 = max1 * max2;
            if (max3 > max)  max = max3;

        }

    cout << max;
    return 0;
}
```

循环太多导致oj超时
但是助教会批改，短暂超时不扣分，除非你的程序陷入死循环，无法运行

助教测试的结果是oj里三重循环都够用

```cpp
#include<iostream>
using namespace std;
int a[1000][1000], l[1000];
int main() {
    int m, n, s, max = 0, x,b;
    cin >> m >> n;
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++) cin >> a[i][j];
    for(int i=0;i<m;i++)
        for (int j = 0; j < n; j++) {
            for (int s = 0; s < m + 10; s++) l[s] = 0;
            x = 0; b = 0;
            if (a[i][j] == 1) {
                for (int e = 0; e < m - i; e++) {
                    if (a[i + e][j] == 1) {
                        while (a[i + e][j + b] == 1 && j + b < n) {
                            l[x] += 1;
                            b += 1;
                        }
                        x += 1;
                    }
                    else break;
                }
            }
            max = l[0];
            for (int s = 1; s < x; s++) {
                if (l[s] != l[s - 1]) {
                    s = l[s - 1] * s;
                    max = max > s ? max : s;
                }
            }
            if (l[x - 1] == l[x - 2]) {
                s = l[x - 1] * x; max = max > s ? max : s;
            }
        }
    cout << max;
    return 0;
}
```

```cpp
#include<iostream>
using namespace std;
int a[101][101];
bool check(int i, int j, int k, int s){
    for(int m = i; m <= k; m++){
        for(int n = j; n <= s; n++)
            if(a[m][n] == 0) return false;
    }
    return true;
}

int main(){
    int n, m;
    int max = 0;
    int temp;
    bool flag = true;
    cin >> n >> m;
    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++){
            cin >> a[i][j];
            if(a[i][j]==1)
                flag = false;
        }
    if(flag){
        cout << 0;
        return 0;
    }
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            for(int k = i; k < m; k++){
                for(int s = j; s < n; s++){
                    if(check(i, j, k, s))
                        temp= (k - i + 1) * (s - j + 1);
                        max =  temp > max ? temp : max;
                }
            }
        }
    }
    cout<<max;
    return 0;
}
```

```cpp
#include<iostream>
using namespace std;
int main()
{
    int n, m;
    cin >> n >> m;
    int* p = new int[n * m];
    for (int i = 0; i < n * m; i++) {
        cin >> p[i];
    }
    int s=1;
    int l;
    int k;
    int d;
    for (int i = 0; i < n * m; i++) {
        if (p[i] == 1) {
            int x = i / m;//行
            int y = i % m;//列
            for (int j = 1; j <= n - x; j++) {
                for (int k = 1; k <= m - y; k++) {
                    int a = 0;
                    for (int b = 0; b < j; b++) {
                        for (int c = 0; c < k; c++) {
                            if (p[i + b * m + c]== 1)
                                a = a + 1;
                            if (a == j * k)
                                d = a;
                            if (d > s)
                                s = d;

                        }
                    }
                }
            }

        }
    }
    cout << s;
    return 0;
}
```

题2：结构体数组合并

创建两个结构体数组**A**、**B**保存学生期末考试信息，合并**A**、**B**并删去相同学号的节点（相同学号的信息一定相同），根据三门课的总分进行由高到低排序并输出所有信息。

结构体模板：

```
struct Student{
        int id;
        char name[20];
        int score[3];
        int all;
};
```

注意读题！

编写并调用函数：

```
int join(Student *A, int N, Student *B, int M, Student *ALL);
void sort(Student *ALL, int S);
void display(Student *ALL, int S);
```

输入：

第一行为两个整数**N**、**M**，分别表示**A**和**B**中的成绩信息数量。

接下来为**N+M**行信息，每行为学号**id**，姓名**name**，成绩**score1 score2 score3**。**name**长度不超过**20**。

输出：

计算**score1-3**的总和并存储在结构内，按由高到低对**A**、**B**合并后的数组进行排序（相同总分则**id**小的排在前列），并输出结构内信息（包括总分）。

样例输入：

2 3
191330025 Peach 90 84 87
191570124 Pineapple 76 75 83
191770220 Coconut 92 94 99
191330025 Peach 90 84 87
182110111 Strawberry 85 96 80

样例输出：

191770220 Coconut 92 94 99 285
182110111 Strawberry 85 96 80 261
191330025 Peach 90 84 87 261
191570124 Pineapple 76 75 83 234

input:

3 4
184590034 Apple 78 72 83
143330123 Pear 79 71 83
191420089 Orange 80 70 83
184590034 Apple 78 72 83
143330123 Pear 79 71 83
191420089 Orange 80 70 83
190120210 Cherry 90 87 84

output:

190120210 Cherry 90 87 84 261
143330123 Pear 79 71 83 233
184590034 Apple 78 72 83 233
191420089 Orange 80 70 83 233

input:

1 1
191330025 Peach 90 84 87
143330123 Pear 79 71 83

output:

191330025 Peach 90 84 87 261
143330123 Pear 79 71 83 233

input:

4 2
184590034 Apple 78 72 83
143330123 Pear 79 71 83
191420089 Orange 80 70 83
190120210 Cherry 90 87 84
191570124 Pineapple 76 75 83
191770220 Coconut 92 94 99

output:

191770220 Coconut 92 94 99 285
190120210 Cherry 90 87 84 261
191570124 Pineapple 76 75 83 234
143330123 Pear 79 71 83 233
184590034 Apple 78 72 83 233
191420089 Orange 80 70 83 233

```cpp
#include <iostream>
using namespace std;
struct Student {
    int id;
    char name[20];
    int score[3];
    int all;
};

int join(Student* A, int N, Student* B, int M, Student* ALL);
void display(Student* ALL, int S);
void sort(Student* ALL, int S);
```

```cpp
int main() {
    int N, M;
    cin >> N >> M;
    Student* A = new Student[N];
    Student* B = new Student[M];
    for (int i = 0; i < N; i++) {
        cin >> A[i].id >> A[i].name;
        A[i].all = 0;
        for (int j = 0; j < 3; j++) {
            cin >> A[i].score[j];
            A[i].all += A[i].score[j];
        }
    }
    for (int i = 0; i < M; i++) {
        cin >> B[i].id >> B[i].name;
        B[i].all = 0;
        for (int j = 0; j < 3; j++) {
            cin >> B[i].score[j];
            B[i].all += B[i].score[j];
        }
    }
    Student* ALL = new Student[N + M];
    int p = join(A, N, B, M, ALL);
    sort(ALL, p);
    display(ALL, p);
    return 0;
}
```

```
42  int join(Student* A, int N, Student* B, int M, Student* ALL) {
43      for (int i = 0; i < N; i++) {
44          ALL[i] = A[i];
45      }
46      int p = N;
47      bool repeat = false;
48      for (int i = 0; i < M; i++) {
49          repeat = false;
50          for (int j = 0; j < N; j++) {
51              if (B[i].id == A[j].id) {
52                  repeat = true;
53                  break;
54              }
55          }
56          if (!repeat) {
57              ALL[p] = B[i];
58              p++;
59          }
60      }
61      return p;
62  }
```

```
64  void sort(Student* ALL, int S) {
65      for (int i = 0; i < S - 1; i++) {
66          for (int j = 0; j < S - 1 - i; j++) {
67              if (ALL[j].all < ALL[j + 1].all) {
68                  Student temp = ALL[j];
69                  ALL[j] = ALL[j + 1];
70                  ALL[j + 1] = temp;
71              }
72              else if (ALL[j].all == ALL[j + 1].all && ALL[j].id > ALL[j + 1].id) {
73                  Student temp = ALL[j];
74                  ALL[j] = ALL[j + 1];
75                  ALL[j + 1] = temp;
76              }
77          }
78      }
79  }
```

```
81  void display(Student* ALL, int S) {
82      for (int i = 0; i < S; i++) {
83          cout << ALL[i].id << " " << ALL[i].name << " ";
84          for (int j = 0; j < 3; j++) {
85              cout << ALL[i].score[j] << " ";
86          }
87          cout << ALL[i].all << endl;
88      }
89  }
```

典型问题

```
student* A[100], *B[100], *ALL[100];


for (int i = 0; i < m; i++) {
    cin >> a >> b >> c[0] >> c[1] >> c[2];
    A[i]->id = a;
    A[i]->name = b;
    A[i]->score[0] = c[0];
    A[i]->score[1] = c[1];
    A[i]->score[2] = c[2];
}
```

创建了指针数组，但是没有给每个指针分配空间。

典型问题

```
ALL->name[20] = A->name[20] = name[20];
```

字符串赋值错误。

这个代表字符赋值，并且溢出了。

```
for (int i = 0; i < N; i++)
{
    cin >> s1.id;
    cin >> s1.name;
    cin >> s1.score[0];
    cin >> s1.score[1];
    cin >> s1.score[2];
    s1.all = s1.score[0] + s1.score[1] + s1.score[2];
}
```

```
A = &s1;
```

循环并没有输入进数组，只是一直在修改s1的值。

其他问题：

1.排序时没有考虑同分数不同id的排名

2.合并A、B表时没有去掉重复

3.字符串初始化、赋值问题（除了过于严重的赋值错误，这里都没有扣分）

评分标准：

Main()函数	1分
Join()函数	3分
Sort()函数	3分
Display()函数	2分

谢　谢！