

每日一题C++方向day24_12月2日测评结果

考生信息



王婧

考号：2378 | 学校：财经大学 | 邮箱：1031160332@qq.com | 职位：54

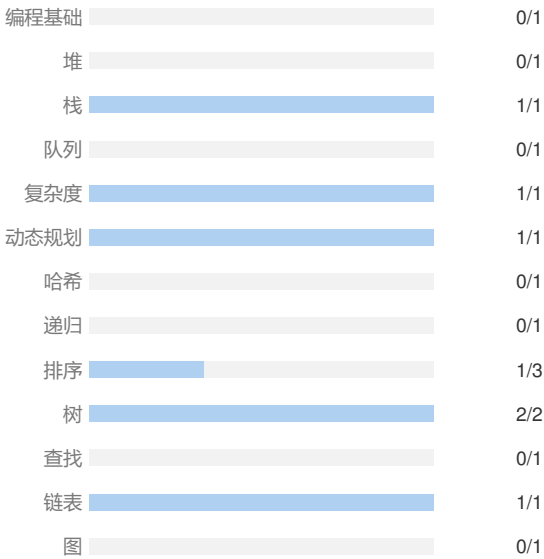
参考区域: 陕西省西安市 (113.139.122.161) | 做题用时：00:56:54(2019-12-02 19:29:11 - 20:26:18) | 作答设备：PC

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	20.0	4	81	00:05:50	--
编程	25.0	1	42	00:47:02	--

知识点技能图谱



知识点	得分	正确题数
编程基础	0.0	0
堆	0.0	0
栈	5.0	1
队列	0.0	0
复杂度	5.0	1
动态规划	25.0	1
哈希	0.0	0
递归	0.0	0
排序	5.0	1
树	10.0	2
查找	0.0	0
链表	5.0	1
图	0.0	0

历史笔试记录

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
1	54班C/C++考试题	11.0%	21.0/60	单选:6.0分 编程:15.0分	否	2019-09-21 17:24:48	2019-09-22 15:38:37
2	54班CPP_DS_2_考试卷	38.0%	19.0/60	单选:4.0分 编程:15.0分	否	2019-10-31 17:40:31	2019-11-02 15:40:22
3	每日一题C++方向day02_11月7日	26.0%	70.0/100	单选:20.0分 编程:50.0分	否	2019-11-06 16:54:27	2019-11-07 13:25:19
4	每日一题C++方向day03_11月8日	16.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-07 14:40:49	2019-11-08 12:51:53
5	每日一题C++方向day04_11月9日	1.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-11-08 11:17:08	2019-11-10 17:00:19
6	每日一题C++方向day05_11月10日	64.0%	52.5/100	单选:25.0分 编程:27.5分	是，相似代码	2019-11-09 15:35:20	2019-11-10 17:49:42
7	每日一题C++方向day06_11月11日	54.000004%	65.0/100	单选:40.0分 编程:25.0分	否	2019-11-09 15:38:30	2019-11-11 18:14:20
8	每日一题C++方向day07_11月13日	50.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-12 11:39:41	2019-11-13 14:37:22
9	每日一题C++方向day08_11月14日	8.0%	90.0/100	单选:40.0分 编程:50.0分	是，相似代码	2019-11-13 10:51:46	2019-11-14 17:35:22
10	每日一题C++方向day09_11月15日	79.0%	45.0/100	单选:20.0分 编程:25.0分	否	2019-11-14 18:14:54	2019-11-15 17:52:51
11	每日一题C++方向day10_11月16日	6.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-15 13:29:43	2019-11-16 12:12:48
12	每日一题C++方向day11_11月17日	22.0%	75.0/100	单选:25.0分 编程:50.0分	否	2019-11-16 10:59:12	2019-11-17 21:38:02
13	每日一题C++方向day12_11月18日	45.0%	70.0/100	单选:20.0分 编程:50.0分	是，相似代码	2019-11-17 11:32:33	2019-11-18 17:49:04
14	每日一题C++方向day13_11月20日	2.0%	90.0/100	单选:40.0分 编程:50.0分	是，相似代码	2019-11-19 11:13:30	2019-11-20 15:08:03
15	每日一题C++方向day14_11月21日	16.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-20 13:55:05	2019-11-21 13:15:24
16	每日一题C++方向day15_11月22日	21.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-21 12:07:32	2019-11-22 12:56:30
17	每日一题C++方向day16_11月23日	47.0%	55.0/100	单选:5.0分 编程:50.0分	是，相似代码	2019-11-22 14:21:38	2019-11-23 12:51:07
18	每日一题C++方向day17_11月24日	2.0%	95.0/100	单选:45.0分 编程:50.0分	否	2019-11-23 14:28:02	2019-11-25 15:04:52
19	每日一题C++方向day18_11月25日	47.0%	70.0/100	单选:20.0分 编程:50.0分	是，相似代码	2019-11-24 11:44:46	2019-11-25 15:42:37
20	每日一题C++方向day19_11月27日	22.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-25 14:38:07	2019-11-27 12:59:39
21	每日一题C++方向day20_11月28日	8.0%	90.0/100	单选:40.0分 编程:50.0分	否	2019-11-25 18:26:05	2019-11-28 16:59:04
22	每日一题C++方向day21_11月29日	22.0%	75.0/100	单选:25.0分 编程:50.0分	是，相似代码	2019-11-27 12:24:51	2019-11-29 17:59:31
23	每日一题C++方向day22_11月30日	21.0%	85.0/100	单选:35.0分 编程:50.0分	否	2019-11-29 11:02:40	2019-11-30 16:08:29
24	每日一题C++方向day23_12月1日	52.999996%	60.0/100	单选:35.0分 编程:25.0分	否	2019-11-29 15:38:28	2019-12-02 19:27:28

编码能力

规范性 (4.9/5) 正确性 (2.5/5) 思路 (3.0/5)

题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题1	100%	10	00:31:03	C++	5ms	492K	良	良	1%

题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题2	0%	3	00:15:59	C++	0ms	0K	差	优	32%

1 [平均分2.4分 | 63人正确/134人做题 | 用时：<1分 | 得分：5.0 / 5.0]

将N条长度均为M的有序链表进行合并，合并以后的链表也保持有序，时间复杂度为()？

- A $O(N * M * \log N)$
- B $O(N * M)$
- C $O(N)$
- D $O(M)$

他的回答：A (正确)

正确答案：A

2 [平均分4.5分 | 120人正确/132人做题 | 用时：<1分 | 得分：5.0 / 5.0]

下设法S的初始状态为空，元素a,b,c,d,e,f依次入栈S，出栈的序列为b,d,c,f,e,a则栈S的容量至少为（ ）

- A 6
- B 5
- C 4
- D 3

他的回答：D (正确)

正确答案：D

3 [平均分2.8分 | 74人正确/133人做题 | 用时：<1分 | 得分：0.0 / 5.0]

大小为MAX的循环队列中，f为当前对头元素位置，r为当前队尾元素位置(最后一个元素的位置)，则任意时刻，队列中的元素个数为

- A r-f
- B $(r-f+MAX+1)\%MAX$
- C $r-f+1$
- D $(r-f+MAX)\%MAX$

他的回答：D (错误)

正确答案：B

4 [平均分3.2分 | 81人正确/128人做题 | 用时：<1分 | 得分：0.0 / 5.0]

n！后面有多少个0, $6! = 1 * 2 * 3 * 4 * 5 * 6 = 720$. 720后面有1个0，n=10000，求n！。

- A 2498
- B 2499
- C 2450
- D 2451

他的回答：C (错误)

正确答案：B

5 [平均分3.3分 | 85人正确/130人做题 | 用时：<1分 | 得分：5.0 / 5.0]

若一棵二叉树具有12个度为2的结点，6个度为1的结点，则度为0的结点个数是（ ）。

- A 10

- B 11
- C 13
- D 不确定

他的回答： C (正确)

正确答案： C

6 [平均分2.3分 | 60人正确/132人做题 | 用时：<1分 | 得分：5.0 / 5.0

若将关键字1, 2, 3, 4, 5, 6, 7 依次插入到初始为空的平衡二叉树 T 中，则 T 中平衡因子为 0 的分支结点的个数是（ ）。

- A 0
- B 1
- C 2
- D 3

他的回答： D (正确)

正确答案： D

7 [平均分2.0分 | 54人正确/132人做题 | 用时：<1分 | 得分：0.0 / 5.0

已知小根堆为8,15,10,21,34,16,12，删除关键字 8 之后需重建堆，在此过程中，关键字之间的比较次数是（ ）。

- A 1
- B 2
- C 3
- D 4

他的回答： B (错误)

正确答案： C

8 [平均分2.9分 | 75人正确/128人做题 | 用时：<1分 | 得分：0.0 / 5.0

已知某个哈希表的n个关键字具有相同的哈希值，如果使用二次探测再散列法将这n个关键字存入哈希表，至少要进行____次探测。

- A n-1
- B n
- C n+1
- D n(n+1)
- E n(n+1)/2
- F 1+n(n+1)/2

他的回答： F (错误)

正确答案： E

9 [平均分2.0分 | 51人正确/128人做题 | 用时：<1分 | 得分：0.0 / 5.0

下列选项中，不可能是快速排序第2趟排序结果的是（ ）

- A 2,3,5,4,6,7,9
- B 2,7,5,6,4,3,9
- C 3,2,5,4,7,6,9
- D 4,2,3,5,7,6,9

他的回答： B (错误)

正确答案： C

10 [平均分1.0分 | 25人正确/129人做题 | 用时：<1分 | 得分：0.0 / 5.0

设有向图G=(V,E)，顶点集 V={V0,V1,V2,V3}，边集 E={<v0,v1>，<v0,v2>，<v0,v3>，<v1,v3>}。若从顶点 V0 开始对图进行深度优先遍历，则可能得到的不同遍历序列个数是（ ）。

- A 2
- B 3
- C 4
- D 5

他的回答： B (错误)

正确答案： D

11 [平均分23.5分 | 97人正确/103人做题 | 提交: 10 次 | 得分：25.0 / 25.0]

标题：年终奖 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【年终奖】

小东所在公司要发年终奖，而小东恰好获得了最高福利，他要在公司年会上参与一个抽奖游戏，游戏在一个6*6的棋盘上进行，上面放着36个价值不等的礼物，每个小的棋盘上面放置着一个礼物，他需要从左上角开始游戏，每次只能向下或者向右移动一步，到达右下角停止，一路上的格子中的礼物小东都能拿到，请设计一个算法使小东拿到价值最高的礼物。

给定一个6*6的矩阵board，其中每个元素为对应格子的礼物价值,左上角为[0,0].请返回能获得的最大价值，保证每个礼物价值大于100小于1000。

输入描述：

输出描述：

代码片段

功能实现	代码提交统计		代码执行统计
TA的 平均	TA的 平均		
总通过率 100%	使用语言 C++	答案错误：1	
基本测试用例通过率 1/1 (100%) 94%	做题用时 00:31:03 00:33:23	段错误：3	
	提交次数 10 4	编译错误：5	
		答案正确：1	
代码效率		代码规范及可读性	
TA的 参考	代码规范得分 4.7619		
运行时间 5ms 3s	Line 12: Add #include for vector<> [build/include_what_you_use] [4]		
占用内存 492K 32768K			

他的代码：

做题用时: 31 分钟 语言：C++ 运行时间：5ms 占用内存：492K 程序状态：答案正确

```
#include<algorithm>
class Bonus {
public:

    int getMost(vector<vector<int> > board) {
        // write code here
        int length=board.size();
        int width=board[0].size();
        vector<vector<int>> price;
        for(int i=0;i<length;i++)
        {
            vector<int> tmp(width,0);
            price.push_back(tmp);
        }
        price[0][0]=board[0][0];
        for(int i=0;i<length;i++)
        {
            for(int j=0;j<width;j++)
            {
                if((i==0&&j==0)
```

```

        {
            continue;
        }
        else if(i==0)
        {
            price[i][j]=price[i][j-1]+board[i][j];
        }
        else if(j==0)
        {
            price[i][j]=price[i-1][j]+board[i][j];
        }
        else
        {
            price[i][j]=max(price[i][j-1],price[i-1][j])+board[i][j];
        }
    }
}
return price[length-1][width-1];
}
};

```

12 [平均分17.5分 | 45人正确/65人做题 | 提交: 3 次 | 得分: 0.0 / 25.0]

标题: 迷宫问题 | 时间限制: 1秒 | 内存限制: 32768K | 语言限制: 不限

【迷宫问题】

定义一个二维数组N*M (其中 $2 \leq N \leq 10$; $2 \leq M \leq 10$) , 如 5×5 数组下所示:

```

int maze[5][5] = {

    0, 1, 0, 0, 0,

    0, 1, 0, 1, 0,

    0, 0, 0, 0, 0,

    0, 1, 1, 1, 0,

    0, 0, 0, 1, 0,

};

```

它表示一个迷宫, 其中的1表示墙壁, 0表示可以走的路, 只能横着走或竖着走, 不能斜着走, 要求程序找出从左上角到右下角的最短路线。入口点为[0,0], 既第一空格是可以走的路。

Input

一个 $N \times M$ 的二维数组, 表示一个迷宫。数据保证有唯一解, 不考虑有多解的情况, 即迷宫只有一条通道。

Output

左上角到右下角的最短路径, 格式如样例所示。

Sample Input

```

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

```

Sample Output

```

(0, 0)
(1, 0)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(2, 4)
(3, 4)
(4, 4)

```

输入描述：

输入两个整数，分别表示二位数组的行数，列数。再输入相应的数组，其中的1表示墙壁，0表示可以走的路。数据保证有唯一解,不考虑有多解的情况，即迷宫只有一条通道。

输出描述：

左上角到右下角的最短路径，格式如样例所示。

示例1：

输入

5 5
0 1 0 0 0
0 1 0 1 0
0 0 0 0 0
0 1 1 1 0
0 0 0 1 0

输出

(0,0)
(1,0)
(2,0)
(2,1)
(2,2)
(2,3)
(2,4)
(3,4)
(4,4)

代码片段

功能实现	代码提交统计			代码执行统计
	TA的	平均	TA的	平均
总通过率	0%	70%	使用语言	C++
基本测试用例通过率	0/6 (0%)	70%	做题用时	00:15:59 00:52:07
边缘测试用例通过率	0/4 (0%)	70%	提交次数	3 4
编译错误	: 3			
代码效率			代码规范及可读性	
TA的	参考		代码规范得分	
运行时间	0ms	1s	5.0	
占用内存	0K	32768K		

他的代码：

做题用时: 15 分钟 语言：C++ 运行时间：0ms 占用内存：0K 程序状态：编译错误

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;
int N,M;
vector<vector<int>> maze;
vector<vector<int>> path_temp;
vector<vector<int>> path_best;
void MazeTrack(int i,int j)
{
    maze[i][j]=1;
```

```

path_temp.push_back({i,j});
if(i==N-1&& j==M-1)
{
    if(path_best.empty()||path_temp.size()<path_best.size())
        path_best=path_temp;

    if(i-1>=0&&maze[i-1][j]==0)
        MazeTrack(i-1,j);
    if(i+1<N&&maze[i+1][j]==0)
        MazeTrack(i+1,j);
    if(j-1>=0&&maze[i][j-1]==0)
        MazeTrack(i,j-1);
    if(j+1<N&&maze[i][j+1]==0)
        MazeTrack(i+1,j);

    maze[i][j]=0;
    path_temp.pop_back();
}
}
int main()
{
    while(cin>>N>>M)
    {
        maze=vector<vector<int>>> (N,vector<int>(M,0));
        path_temp.clear();
        path_best.clear();
        for(auto &i:maze)
            for(auto &j:i)
                cin>>j;

        MazeTrack(0,0);
        for(auto i:path_best)
            cout<<'('<<i[0]<<','<<i[1]<<')'<<endl;
    }
    return 0;
}

```