白熊求职
www.icebear.me

多益网络 2017 春招两个编程题

原文地址:http://blog.csdn.net/KgdYsg/article/details/64494166

题目 1 舍友做的，做完后我听了他的描述，自己写了一个解题程序也不知道对不对，毕竟不是在线 oj 不能查结果。

题目描述，给 n,m 两个数，n 表示球队的个数（从 1 到 n），m 表示比赛场数，然后给 m 组数据，每组数据有两个数，（a,b）即表示 a 赢了 b，要求是输出一种排列方式，使得不存在排在前面的队输过排在后面的队。如果有（a,b）即 a 一定在 b 前面。

我的想法是将这 m 场比赛视为 m 个不同的规则，初始化一个 count 数组，长度为 n，默认排列顺序为（1…n）然后进行 m 次检验，如果（a,b）在现有的排列中不满足 a 在 b 的前面，则调换 a,b 的位置，否则不做处理。

我试了下这个方法，发现有的数据不好使。取 n=5，m=5 五场比赛（2，3）（2，4）（1，2）（1，5）（5，2）

排列的结果：

1 2 3 4 5 比赛（2，3）满足顺序

1 2 3 4 5 比赛（2，4）满足顺序

1 2 3 4 5 比赛（1，2）满足顺序

1 2 3 4 5 比赛（1，5）满足顺序

1 5 3 4 2 比赛（5，2）调整 2，5 位置 但是在这个地方会使 2 排在 3 4

微信关注：白熊事务所，获取更多资料礼包！

后面，以至于不满足前面的规则。

随后进行了改进，把这个循环写成一个函数，sort 进行排序，排序完后进行 test，检查现在的结果是否满足要求，如果不满足，再进行一次 sort，若干次排序后一定会得到一个结果满足条件。

代码如下：

```
#include <iostream>
using namespace std;
void soft(int count[],int a[][2], int m, int n)
{
    for(int i = 0; i < m; i++)
    {
        int j,k,temp;
        for(j = 0; j < n; j++)
        {
            if(count[j] == a[i][0])
            {
                break;
            }
        }
        for(k = 0; k < n; k++)
        {
            if(count[k] == a[i][1])
```

微信关注：白熊事务所，获取更多资料礼包！

白熊求职
www.icebear.me

```
                {
                    break;
                }
            }
            if(j > k)
            {
                temp = count[j];
                count[j] = count[k];
                count[k] = temp;
            }
        }
    }

    bool test(int count[], int a[][2], int m, int n)
    {
        for(int i = 0; i < m; i++)
        {
            int j,k;
            for(j = 0; j < n; j++)
            {
                if(count[j] == a[i][0])
                {
```

微信关注：白熊事务所，获取更多资料礼包！

白熊求职
www.icebear.me

```
                break;
            }
        }
        for(k = 0; k < m; k++)
        {
            if(count[k] == a[i][1])
            {
                break;
            }
        }
        if(j > k)
        {
            return false;
        }
    }
    return true;
}
int main()
{
    int n,m;
    cin>>n>>m;
    int a[m][2];
```

微信关注：白熊事务所，获取更多资料礼包！

```
        for(int i = 0; i < m ; i++)
        {
            cin>>a[i][0]>>a[i][1];
        }
        int count[n];
        for(int i = 0; i < n; i++)
        {
            count[i] = i+1;
        }
        while(!test(count,a,m,n))
        {
            soft(count,a,m,n);
        }
        for(int i = 0; i < n; i++)
        {
            cout<<count[i]<<" ";
        }
        return 0;
}
```

可能还是会有些情况没有考虑到，但是因为这个不是提交就能看结果的测评，所以没办法去验证。

微信关注：白熊事务所，获取更多资料礼包！

题目 2 给一个数组 a={8，3，5，2，7，4，6，1}，要求用 O(n)的时间复杂度，O(1)的空间复杂度做。

做法：申请一个 count 数组（所有元素为 0），保证数组的长度大于要排序的数组的最大元素，然后遍历数组 a，把 a 中每一个元素的值做为 count 数组的下标，并使该下标对应的数组元素的值自增。用空间换取时间的非典型排序方法。

代码如下：

```
#include <iostream>
using namespace std;
void sort(int count[], int a[], int len)
{
    for(int i = 0; i < len; i++)
    {
        count[a[i]]++;
    }
    for(int i = 0, j = 0; i < 65536; i++)
    {
        if(count[i] != 0)
        {
```

微信关注：白熊事务所，获取更多资料礼包！

```
                a[j++] = i;
                count[i]--;
                i--;
            }
        }
    }
    void show(int a[], int len)
    {
        for(int i = 0; i < len; i++)
        {
            cout<<a[i]<<" ";
        }
    }
    using namespace std;
    int main()
    {
        int count[65536] = {0};
        int a[] = {8,3,5,2,7,4,6,1};
        int len = sizeof(a)/sizeof(int);
        sort(count,a,len);
        show(a,len);
        return 0;
```

微信关注：白熊事务所，获取更多资料礼包！

白熊求职
www.icebear.me

```
}
```

icebear.me

白熊事务所致力为准备求职的小伙伴提供优质的资料礼包和高效的求职工具。礼包包括**互联网、金融等行业的求职攻略**；**PPT模板**；**PS技巧**；**考研资料**等。

微信扫码关注：**白熊事务所**，获取更多资料礼包。

登陆官网：**www.icebear.me**，教你如何**一键搞定名企网申**。