

C++方向编程题答案

第六周

day35

1、题目ID: 772 年会抽奖

链接: <https://www.nowcoder.com/questionTerminal/610e6c0387a0401fb96675f58cda8559>

【题目解析】:

这是一道典型的[错排问题](#)，是算法的基本模型。

【解题思路】:

错排的递推公式是: $D(n) = (n - 1) [D(n - 2) + D(n - 1)]$ ，也就是 $n - 1$ 倍的前两项和。公式推导可以参考百度百科。通过这个递推公式可以得到在总数为 n 的时候，错排的可能性一共有多少种。那么要求错排的概率，我们还需要另一个数值，就是当总数为 n 的时候，所有的排列组合一共有多少种，那么根据排列组合，肯定使用

$$A_n^n$$

的公式来求，也就是 n 的阶乘。所以结果很简单，就是用公式求出第 n 项的错排种类，和 n 的阶乘，然后两者一除，就是概率了。

【示例代码】:

```
#include <iostream>
#include <cstdio>

int main()
{
    long long d[22] = { 0, 0, 1 }; // 错排数量, 预留第一项为0, 配合下文中输入的n
    long long f[22] = { 0, 1, 2 }; // 阶乘
    for (int i = 3; i < 22; i++)
    {
        d[i] = (i - 1) * (d[i - 1] + d[i - 2]); // 错排的递推公式
        f[i] = i * f[i - 1]; // 阶乘的递推公式
    }

    int n;
    while (std::cin >> n)
    {
        printf("%.2f%%\n", 100.0 * d[n] / f[n]); // 用100.0来把结果处理成double, 保留两位小数
    }
    return 0;
}
```

2、题目ID: 840 抄送列表

链接: <https://www.nowcoder.com/questionTerminal/286af664b17243deb745f69138f8a800>

【题目解析】:

本题是在第一行的人名中，查找第二行的人名是否存在。牵涉一个全字匹配的问题。

【解题思路】：

本题最简单的思路，就是把第一行中的每一个人名字都存在一个string容器里，然后和第二行的这个字符串一个一个比过去，看有没有重合的即可。比较简单的思路是可以做一个string的数组，分割完了一比就行了。例如我下文中使用的是比较基础的vector。当然也可以选择一些其他的更方便的容器。需要注意的点是双引号需要特殊处理，别的都是正常的字符串处理。

【示例代码】：

```
#include <iostream>
#include <cstdio>
#include <vector>
#include <algorithm>
#include <string>

int main()
{
    std::string line;

    while (std::getline(std::cin, line)) //第一次的输入抓到line里
    {
        std::vector<std::string> names; //names里存放所有的数据
        size_t pos = 0;

        while (pos < line.length()) //位置如果大于等于了长度，那么证明找完了，双引号的情况从这里跳出
        {
            if (line[pos] == '\\') //找到双引号要处理双引号内部
            {
                size_t end = line.find("\\", pos + 1);
                names.push_back(line.substr(pos + 1, end - pos - 1)); //截取出来放到names里
                pos = end + 2; //跳掉后面的双引号和逗号
            }
            else
            {
                size_t end = line.find(",", pos + 1);

                if (end == -1) //找不到逗号证明这是最后一个字符串了，直接存了这个字符串后跳出
                {
                    names.push_back(line.substr(pos, line.size() - pos));
                    break;
                }
                names.push_back(line.substr(pos, end - pos));
                pos = end + 1; //跳掉后面的逗号
            }
        }

        std::getline(std::cin, line);
        if (names.end() == std::find(names.begin(), names.end(), line))
            //std::find函数是在两个迭代器中间查找，最后一个参数是要找的值。
            //找到了返回对应迭代器，找不到就返回区间末尾的位置。本例中是end()。
        {
            printf("Important!\n"); //没找到
        }
        else
        {
            printf("Ignore\n"); //找到了
        }
    }
    return 0;
}
```

比特科技