ACM/ICPC Beijing Regional 2002 Analysis

Author: chenyan

Problem A: Moving Object Recognition

【题目简述】给定一系列关于一个物体运动的等时间间隔的图像，求出物体的运动速度。
【分析】简单的模拟题。对每幅图用“种子染色法”找出最大的连续区域，该区域就是所求物体，纪录下物体的重心位置。最后用逐差法算出速度。

Problem B: Random Walk

【题目简述】给定一个由若干随机变量组成的程序，程序中只有顺序执行语句 NOP 和条件判断语句 IF，求出每个函数的平均运行时间。
【分析】本题粗看起来是一道模拟题。但由于涉及到随机变量的分析，如果采用模拟随机变量多次运行的方法执行起来效率不高，且难以保证结果的精度。由于程序中没有循环的结构，且涉及到随机变量的命令只有 IF 语句。设每个函数的平均执行时间为 X_i 很容易列出它的方程。以样例数据为例：

```
PROC A
IF x>0.5 GOTO 3;
NOP;
END;
PROC B
IF x<0.5 PROC A;
NOP;
END;
```

对于函数 A，由于条件转移的条件是 X>0.5，每个分支被执行的概率是 0.5，因此运行时间是 $X_1=1*0.5+2*0.5$。
对于函数 B，类似的，运行时间是 $X_2=(X_1+2)*0.5+2*0.5$，虽然这里 X_1 的值不知道，但通过上面的方程 $X_1=1*0.5+2*0.5$ 可以解出，将其带入 X_2 可以得到最终结果。

由此容易发现本题的求解过程实际上就是求一个线性方程组的解。用高斯消元法即可。

【推广】1、考虑有循环和嵌套的情况下，如何求平均运行时间。2、参见 zju1198:Single-Player Games。

Problem C: Radar Installation

【题目简述】给定点集 S={(x,y)|y>0}，求用圆心在 x 轴上，半径为 d 的圆覆盖 S 所需的最少圆个数。
【分析】依题意，S 中的每个点 (x,y) 可以对应为 x 轴上的一个闭区间 $\left[x-\sqrt{d^2-y^2}, x+\sqrt{d^2-y^2}\right]$，当圆心位于该区间内均可覆盖到点 (x,y)。若该区间不存在则问题无解。容易证明本题可以用贪心算法来解决。

```
1：将区间集 Seg 以右端点为关键字从小到大排序
2：for i=1 to n
3：  if seg[i]未标记 then
4：    ans=ans+1; 标记 seg[i]
5：    for j=i+1 to n
6：      if seg[j]左端点在 seg[i]的右端点之前 且 seg[j]未标记 then 标记 seg[j]
```

7：return ans;

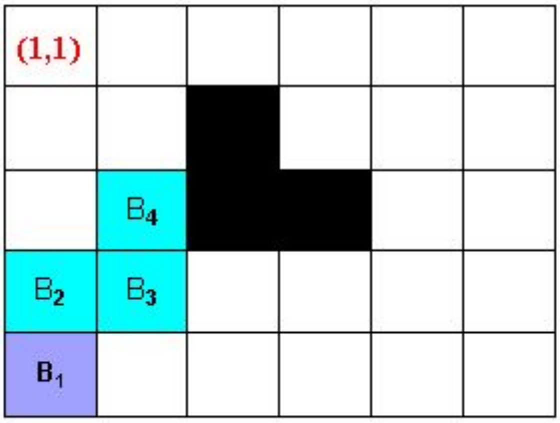
另外，由于本题涉及到浮点运算，判断大小时不能直接使用==,<,>判断。

Problem D:Holedox

【题目简述】求一条长度为 L 的蛇在迷宫中走到出口的最少步数。

【分析】本题属于运动规划类问题，运动物体的形状不确定属于 NP 问题，因此考虑采用广度优先搜索。

【状态表示】蛇头的坐标(x,y)，蛇的形状 shape。规定四个方向分别对应 0,1,2,3。由于蛇的形状是连续的，我们可以记录下蛇的每部分相对于上一部分的方向即可得到形状。考虑到空间限制，可以采用按位存储，每个部分只要两个 2 进制位，这样蛇的形状用一个整形变量即可存下。



常量：

int dir[4][2]={{0,1},{1,0},{0,-1},{-1,0}};

L=4

当前状态：

x=5; y=1;

shape=$(11\ 01\ 11)_2$

【判重】采用 Hash 表。以三元组(shape,x,y)作为 Hash 函数。

Problem E: Game Prediction

【题目简述】M*N 张点数不同的牌分给 M 个人，每轮中出牌点数最大的人获胜。给定一个游戏者的牌，确定其必胜场数。

【分析】本题是一道简单的模拟题。算法如下：

1：将牌从大到小排列

2：ans=0;

2：对于每张牌：

3：　如果对手有更大的牌，则出最大的一张牌，其余的人出最小的牌。

4：　如果对手没有更大的牌，则 m-1 个人均出最小的牌且 ans++;

Problem F: Chocolate

【题目简述】一个口袋中有 n 个球，球的颜色有 c 种。现从口袋中取出一个球。若取出的球与桌上已有球颜色相同，则将两球都取走，否则将取出的球放在桌上。设从口袋中取出每种颜色的球的概率均等。求取出 n 个球后桌面上剩余 m 个球的概率。

【分析】显然本题的状态为二元组(n,m)。要达到(n,m)状态，只有两种方法：

1、取出的球与桌面上已有的 m+1 个球中的 1 个同色，此时可通过取走这个球实现(n-1,m+1)→(n,m)的转化，此事件发生的概率为(m+1)/c。

2、取出的球与桌面上已有的 m-1 个球均不同色，此时可通过将取出的球放在桌上实现(n-1,m-1)→(n,m)的转化，此事件发生的概率为 1-(m-1)/c。

由此可得本题的递推方程：

$$f(n,m)=\frac{m+1}{c}f(n-1,m+1)+\frac{c-(m-1)}{c}f(n-1,m-1)$$

边界条件：显然 f(0,0)=1, f(1,1)=1, f(n,m)=0 (m>c), f(0,m)=0 (m<>0)

编程时可以使用取余法减少存储空间，但由于取余运算时间复杂度比较高，可以将 f[n%2][m]改为 f[n&1][m]。但这样仍要计算(n-1)&1 的值，也浪费了时间。可以设一个变量 cur，每次外层循环时 cur=1-cur，这样上一阶段就是 1-cur。优化后对 n=10^6 的数据可节省 1 秒。

但这样优化后仍然超时。这时就需要对问题的数学性质进行分析。通常有两条路：1、通过母函数理论将递推关系转化为“封闭”的代数式，这种做法的效率高，但需要较多的数学知识；2、通过数论、不等式、随机化等手段，简化计算。

我们从奇偶性入手，观察 m 的值在每个阶段的分布情况：n=1 时 m={1};n=2 时 m={0,2};n=3 时 m={1,3};n=4 时 m={0,2,4}……

由此易得，当(n-m)%2==1 时 f(n,m)=0。进一步地，实际计算时我们只需计算 n&1 至 c 中与 n 奇偶性相同的项即可。这样计算量就减少了一半。从空间上分析，由于本题的特殊奇偶性规律，数据结构只要用 1 维数组就足够了，这样做同时也减少了数组寻址的额外浪费。

当然，对于 c,n 相同的测试数据，只需算一次就可以了。

优化至此，已能在考试的时限(10s)内出解，但效率还不理想。考虑到本题精度要求不高，只有 3 为有效数字，当 n 较大时对结果并无影响，因此对于 n>10000 的情形，可以令 n=10000（n 为偶数）或 n=9999（n 为奇数）这样本题的效率将大大提高，可在 0.1s 内出解。

【总结】本题虽为一道简单的动态规划试题，但在细节上有许多可以优化之处，特别是最后一步优化，对效率的提高其了决定性的作用，这就需要我们在比赛时能随机应变。

Problem G: Machine Schedule

【题目简述】给定 n 项独立的工作，每项工作可以在 A 机器的 Xi 模式上完成；或在 B 机器的 Yi 模式上完成。要求适当安排工作的执行顺序和使用的机器，使两台机器转换模式的总数最小。

【分析】由于工作总数较多(n>1000)搜索显然无法胜任。我们尝试建立图论模型来解决本题。如果将任务作为顶点，则不容易找出模式转换数的关系。因此我们以机器的工作模式为顶点，分为两个集合 A,B。分别表示两个机器的工作模式。如果某个任务能够在 A 的 x 模式和 B 的 y 模式下完成，则从 x 到 y 连接一条边，得到一个二部图。原问题就是求这个二部图的最小顶点覆盖数目。

根据 **Koning 定理：二部图的最小顶点覆盖数目等于其最大匹配数目。**可以知道本题实际上就是求二部图的最大匹配，用匈牙利算法即可。

Problem H: Mileage Bank

送分题，没什么可说的。