

ZOJ Monthly, February 2009 解题报告

watashi@zju*

February 14, 2008

Contents

A Cut The Cake	3
B Pie Bridge	4
C Couples	5
D Damn Couples	6
E To Go or Not to Go	7
F Apple Pile	8
G Cookie Choice	9
H Party of 8g	11
I Lazy Tourist	12

*Zejun.WU@gmail.com

ZOJ Monthly, February 2009

Problems of Valentine's Day

Description: ZOJ Monthly, February 2009 – Problems of Valentine's Day

Start Time: 2009-02-15 12:00:00 (GMT+8)

Length: 5 Hours

ID	Title	Ratio (AC/All)	Tag
A	Cut The Cake	22.31% (160/717)	Brute Force & Search
B	Pie Bridge	19.40% (13/67)	Bisection & Search
C	Couples	19.90% (84/422)	Dynamic Programming
D	Damn Couples	37.50% (15/40)	Dynamic Programming
E	To Go or Not to Go	20.27% (60/296)	Counting
F	Apple Pile	61.36% (27/44)	Greedy
G	Cookie Choice	16.66% (3/18)	Dynamic Programming
H	Party of 8g	18.40% (23/125)	Network Flows
I	Lazy Tourist	20.12% (66/328)	Graph Theory

A Cut The Cake

3158 Cut The Cake

Time Limit: 1 Second Memory Limit: 32768 KB

Author: HE, Xing

李雷和韩梅梅搞暧昧情人节分蛋糕吃，蛋糕由 $m*n$ 单元小块组成，每小块包含一定的营养价值。现在要求将蛋糕延纵向分成锯齿形的两块，要求两块的营养价值之差尽量地小。如果答案大于某给定值 t ，则输出“You'd better buy another one!”。

对于每一行，都要从某一位置分成两段，由于不能切出0长度的一段，所以有 $n - 1$ 个选择，一共 m 行，就是 $(n - 1)^m$ 种方案。由于问题的规模很小，暴力枚举所有情况取最优解就可以了。通过预处理和递归可以实现时间复杂度为 n^m 的程序。

B Pie Bridge

3159 Pie Bridge

Time Limit: 10 Seconds Memory Limit: 32768 KB

Author: HANG, Hang

情人节月老为了让喜鹊组成鹊桥要让大家聚在同一地点开会。假想所有喜鹊都在一个二维平面内，且喜鹊在x方向和y方向各有一个速度上限，而月老一次只能通知一只喜鹊到指定地点集合，每两次间隔时间 t ，要求聚集所有喜鹊用时最短。

显然问题关于时间是满足可二分性的，而确定了时间后，假如再确定了月老催每只喜鹊的顺序，那么喜鹊所能到达的区域是一个可简单计算确定的边平行于坐标轴的矩形。对于 N 个这样的矩形判断其是否有公共部分只要简单的坐标比较就可以知道了。而公共部分内的点都可以作为集结点。所以只要取合适的上下界，二分时间，枚举所有排列，再判断是否有公共部分，然后继续二分，直到满足精度要求为止，输出答案就可以了。

假设二分的时间为 T ，月老计划第 k 个催喜鹊 i ，则这只喜鹊对应的矩形为

$$[x_i - v_x \bar{t}, x_i + v_x \bar{t}] \times [y_i - v_y \bar{t}, y_i + v_y \bar{t}], \quad \bar{t} = \max \{T - (k - 1)t, 0\}.$$

如果有

$$\max x_{min} \leq \min x_{max}, \max y_{min} \leq \min y_{max}$$

那么所有矩形交集非空。

对于枚举所有顺序，C++可以直接利用std::next_permutation，而如果自己实现的话可以合并公共运算，提前剪枝，速度应该会更好。

C Couples

3160 Couples

Time Limit: 1 Second Memory Limit: 32768 KB

Author: ZHUANG, Junyuan

不超过300个人坐成一排，如果相邻两人有八卦关系，那么这两个人可以私奔，问最优方案下可以有多少人私奔。

这是一个很经典¹的动态规划问题。注意初始状态是0 $n - 1$ 的一个排列，需要处理一下。

记 $f[l][r]$ 表示从 l 到 r 这一段的人是否可以集体私奔。那么 $f[l][r]$ 为真有两种情况：

- 位置 l 和 r 的人有八卦关系且 $f[l + 1][r - 1]$ 为真；
- 存在某个 m 使得 $f[l][m]$ 和 $f[m + 1][r]$ 均为真。

于是可以动态规划求出所有的 $f[i][j]$ ，复杂度为 $O(n^3)$ 。

然后记 $g[n]$ 为前 n 人最少留下的人数，那么有

$$\begin{cases} g[0] &= 0 \\ g[i] &= \min \{g[i - 1] + 1, \min_{j < i, f[j][i]=T} g[j]\} \end{cases}$$

这一段复杂度为 $O(n^2)$ ，答案就是 $n - dp[n]$ 。

¹当我们说一个东西经典的时候，也可以换一种说法，tooold？

D Damn Couples

3161 Damn Couples

Time Limit: 1 Second Memory Limit: 32768 KB

Author: WANG, Yuting

传说中的“去死去死团”有 n 个团员，坐成一排，还有一个非常无聊的团长叫Wyest。Wyest团长非常无聊，于是就发明了一个游戏，先列出了一些八卦，然后选择某个顺序每分钟报一个八卦，如果八卦中的两个人相邻，那么其中一人就会假装纯洁离开。团长真的是非常非常无聊，于是就想走的人越少越好，而团员们想走的人越多越好，问最后最多能留多少人。

显然Wyest可以先贪心地把那些原本不相邻的八卦报掉，这样没有人会走，接下来的所有人一定是由两头的人与外人没有八卦，而内部相邻两人之间有八卦的几段人组成的。要求能剩下最多的人，就是每一段都剩下尽可能多的人，而这个问题可以动态规划预处理。

记 $dp[n]$ 为有 n 个人的这样一段最多能留下的人数，那么有

GAO()

```
1   $dp[0] \leftarrow 0$ 
2  for  $i \leftarrow 1$  to  $maxn$ 
3      do  $dp[i] \leftarrow \infty$ 
4           $\triangleright$  枚举报第 $j$ 和 $j+1$ 人之间的八卦
5          for  $j \leftarrow 0$  to  $i - 1$ 
6              do
7                   $\triangleright$  团员们想走的人越多越好
8                   $tmp \leftarrow \max(dp[\max(0, j - 1)] + dp[\max(0, i - 1 - j)],$ 
9                       $dp[\max(0, j)] + dp[\max(0, i - 2 - j)])$ 
10                  $\triangleright$  团长想走的人越少越好
10                  $dp[i] \leftarrow \min(dp[i], tmp)$ 
```

这个预处理的复杂度为 $O(maxn^2)$ 。

顺便一提传说中的“3○定律”，也就是 $dp[n]$ 有 $dp[n] = n - n/3$ 的性质。证明可通过分类讨论得出。不过这题的规模不需要这个结论。

E To Go or Not to Go

3162 To Go or Not to Go

Time Limit: 1 Second Memory Limit: 32768 KB

Author: WANG, Yuting

给定一个区间 $[a, b]$ ，然后在该区间内等概率地选取一个数 x ，然后随机选取 x 二进制表示中的一位，问取得的是1的概率。

对于区间 $[a, b]$ 的计数问题，我们常常可以转为求区间 $[0, a)$ 和 $[0, b + 1)$ 的求解，这样可以简化问题和代码。而对于区间 $[0, x)$ 的概率，可以用以下算法求得：

CALC-PROBILITY(x)

```
1  ans ← 0
2  len ← BIT-LENGTH( $x$ )
3  ▷ 计算区间 $[0, 2^{len-1})$ 的概率
4   $p[1] \leftarrow 0.5$ 
5  for  $i \leftarrow 2$  to  $len - 1$ 
6      do  $p[i] \leftarrow ((i - 1)/2.0 + 1)/i$           ▷ 区间 $[2^{i-1}, 2^i)$ 的概率
7       $p[i] \leftarrow (p[i - 1] + p[i])/2$ 
8   $ans \leftarrow ans + i^{len-1} * plen - 1$ 
9  ▷ 计算区间 $[2^{len-1}, x)$ 的概率
10  $k \leftarrow 1$ 
11 for  $i \leftarrow len - 2$  downto 0
12     do if TEST-BIT( $x, i$ )          ▷  $\bar{x} = x/2^i * 2^i$ 
13         then  $ans \leftarrow ans + 2^i * (i/2.0 + k)/len$           ▷ 区间 $[\bar{x}, \bar{x} + 2^i)$ 的概率
14          $k \leftarrow k + 1$ 
15  $ans \leftarrow ans/x$ 
16 return ans
```

于是答案就是：

$$\frac{(b + 1)\text{CALC-PROBILITY}(b + 1) - a\text{CALC-PROBILITY}(a)}{b - a + 1}.$$

F Apple Pile

3163 Apple Pile

Time Limit: 1 Second Memory Limit: 32768 KB

Author: WANG, Yuting

一堆苹果摆成边长为 n 个苹果的正四面体，其中一个角的苹果坏了，Wyest每天吃一个好苹果，而到第二天，与坏苹果接触的苹果都会变坏。问Wyest最多能吃多少苹果。

这题迟迟没人AC，和集训当时的情况一样，很多人都没发现这道题是“苹果题”的本质。其实正确的解法就是输入 n ，输出 $n - 1$ ，至于 x 和 y ，保证了都是正数，无视就好了。如果有0倒要特殊处理。

分析：题目中苹果只有好坏是重要的，坏了多少并没影响。显然只要任意选择与坏苹果相对的面上的 $n - 1$ 个苹果，每天吃1个就可以吃 $n - 1$ 个，因为这些苹果要在第 n 天才会坏掉。而到了第 n 天所有的苹果都坏了，所以最多只能吃 $n - 1$ 个苹果。

G Cookie Choice

3164 Cookie Choice

Time Limit: 2 Seconds Memory Limit: 32768 KB

Author: CUI, Tianyi

DD给MM买饼干，每种饼干都有价格和价值，DD身上钞票有限，但他希望给MM买价值尽可能高的饼干。不过MM对饼干是有要求的，首先是对每种饼干的数量上限有要求，然后是MM可能最多要相同组中的一种饼干。没有一种饼干属于同一组。

题目是一个背包问题，不过加了各种限制。

初始设

$$\begin{cases} dp[0] = 0 \\ dp[i] = -\infty \quad i \geq 1 \end{cases}$$

对于分组的情况，我们可以枚举买何种饼干和买的个数，并把花费 p 时，对应所能得到的最大价值记在数组 tp 里。再利用

```
for (int i = d; i >= 0; i--) {
    for (int j = d; j > i; j--) {
        dp[j] = max(dp[j], dp[i] + tp[j - i]);
    }
}
```

更新 dp 数组，约 1000^2 次循环，但是最多只要进行8次。

对于不在任何组的饼干，一种一种处理，处理上限为1和上限为 ∞ 的情况都比较简单。

```
void dp_an_element(int e, int p)
{
    for (int i = d - p; i >= 0; i--) {
        dp[i + p] = max(dp[i + p], dp[i] + e);
    }
}

void dp_inf_element(int e, int p)
{
    for (int i = p; i <= d; i++) {
        dp[i] = max(dp[i], dp[i - p] + e);
    }
}
```

而处理上限 k 既非1也不是 ∞ 的情况的一种方法是引进价格和价值依次为 (p, e) , $(2p, 2e)$, $(2^2p, 2^2e)$, ..., 最后一项使总价格和总价值为 kp 和 ke ，用这共 $\hat{n} = \lceil \lg k \rceil$ 个新物品代替原来物品，每种物品的上限为1，则转化后问题等价。

```
void dp_k_element(int k, int e, int p)
{
    for (int i = 1; i <= k; i <= 1) {
        dp_an_element(e * i, p * i);
        k -= i;
    }
    if (k > 0) {
        dp_an_element(e * k, p * k);
    }
}
```

更好的办法是开一个数组纪录当前状态最少已使用或最多可使用的物体数，动态规划时判断是否可以转移（代码略）。

最后 $dp[d]$ 就是答案，时间规模大约是 10^6 数量级。

H Party of 8g

3165 Party of 8g

Time Limit: 5 Seconds Memory Limit: 56797 KB

Author: CUI, Tianyi

情人节DD要办个party，要求从M男N女中选出若干人，在保证没有男女之间有八卦的前提下使得总权值最大。要求输出最大权值和方案。

其实这是一个最小割模型。构图如下：原点向每个男人连一条流量为对应权值的边，每个女人向汇点连一条流量为对应权值的边，然后对于每组八卦连一条由男到女流量为 ∞ 的边。然后求最小割，那么S集中的男人和T集中的女人就够成一组最优解。

由于八卦关系对应于流量为 ∞ 的边，因此不可能属于割，所以S集中的男人和T集中的女人之接不会有八卦关系。而每一个可行解也对应于把不参与party的人与源或汇的边作为割。而我们求的是最小割，也就是使得不去参加人的总权值最小的解，也就是所要求的去参加的人的总权值最大的解。

I Lazy Tourist

3166 Lazy Tourist

Time Limit: 1 Second Memory Limit: 32768 KB

Author: YANG, Kete

杭州可以看成顶点是旅游经典的有向图，其中一些景点有宾馆，情人节Murphy陪mm出去玩希望出了宾馆之后能最快地回去，问他应该选择住在那里的宾馆。如果有多个最优解，选择编号最小的；如果没有解，输出“I will never go to that city!”。

题目就是要求一个有向图中从有宾馆景点出发的最小环，这只要把所有的 $d(i, i)$ 初始化为 ∞ ，经过floyd算法后，找出景点内有宾馆的最小的 $d(i, i)$ 即可。