动态规划训练赛 1

竞赛时间：2017 年 7 月 21 日 13:00-16:00

题目名称	邮局选址	补兵	任务
输入文件名	post.in	cs.in	mission.in
输出文件名	post.out	cs.out	mission.out
每个测试点时限	1 sec	2 sec	1 sec
内存限制	128M	128M	128M
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	无
题目类型	传统	传统	传统

提交源程序须加后缀

对于 Pascal 语言	post.pas	cs.pas	mission.pas
对于 C 语言	post.c	cs.c	mission.c
对于 C++ 语言	post.cpp	cs.cpp	mission.cpp

注意：最终测试时，所有编译命令均不打开。评测系统为 Win7。

邮局选址

【问题描述】

有 n 个村庄分布在一条直线上，每个村庄可以用一个坐标 x_i 来进行描述。现在，你需要建设 m 个邮局，使得每个村庄到离它最近的邮局的距离之和最小。

【数据规模和约定】

对于100% 的数据，满足 $1 \le n \le 300$，$1 \le m \le 30$，$1 \le x_i \le 10000$。

【简要题解】

首先，如果只能放一个邮局，显然放在中位数最优。

如果放多个邮局，肯定是将村庄按顺序拆成若干段，每段取中位数。于是可以进行 DP：设 $f[i][j]$ 表示到第 i 个村庄已经放了 j 个邮局的最小距离和。预处理出 $g[i][j]$ 表示从 i 到 j 的村庄用一个邮局的最小值后，可以列出方程：

$$f[i][j] = min\{f[k-1][j-1] + g[k][i]\}$$

补兵

【问题描述】

对于一名$DotA$玩家，补兵的个数($creep\ score$)是衡量一名选手的能力的一项重要指标。特别是在打路人局的时候，补兵的能力就更加关键，因为常常会有队友和你抢补刀。比如，队友操控的老鹿开着大在收一波兵，你操作的英雄如果是幽鬼，就需要在老鹿的AOE中偷偷补上几刀来保证自己的发育。

我们可以建立如下模型来大致模拟这种状况：现在有n个小兵，每个小兵都有自己的血量（血量一定是正整数）。你和老鹿轮流对小兵进行攻击。每次，你可以选择对某个小兵造成1点伤害（或者你可以选择不造成任何伤害）；接着，老鹿会对所有小兵都造成1点伤害。如此往复，直到所有小兵都死亡（血量达到0）。在这一过程中，如果你对某个小兵造成了致命伤害（使它的血量由1变为0），那么你就算成功补到这个兵。

对于给定的情形，你需要计算你的最大补兵数。

【数据规模和约定】

对于30%的数据，满足$n \leq 100$。

对于50%的数据，满足$n \leq 500$，$T \leq 30$，每个小兵的血量不超过500。

对于100%的数据，满足$1 \leq n \leq 1000$，$1 \leq T \leq 70$。

【简要题解】

考虑一个最特殊的例子：小兵的血量分布恰好长成1,2,3, ...。如果是这样，我们显然可以补到每一刀。

现在的状况是，我们面对的局面并不是这样，所以我们应该先攻击一些小兵，使得小兵的血量分布变成上面描述的情况（有一个专门的术语就描述了这种操作，叫做“垫刀”）。

还可以明确的情况是，如果有两个小兵的血量一直是一样的，那么这两个小兵我们至多只能补到一个。所以，如果我们发现在血量i没有别的小兵，应该从血量高于i的小兵中，挑一个血量有和别的小兵相同的、血量尽量少的来补齐。这样，我们可以当作这个小兵的血量就是i，只是在选择它前需要垫几刀。

具体到DP，设$f_{i,j}$表示考虑到血量为i的小兵，在操作的过程中省下了j刀的情况下所能补到的最大兵数。如果第i个小兵不补，则$f_{i,j} = f_{i-1,j-1}$；如果要补，则$f_{i,j} = f_{i-1,j+c[i]} + 1$。其中$c[i]$表示要补血量被当作$i$的小兵之前要垫多少刀。两者取一个最大值即可。转移的一些细节可以参考标程。

任务

【问题描述】

n个人正在完成m件任务。

每个人都有一定的能力，我们可以用正整数a_i来描述每个人的能力；每件任务都有一定的难度，我们可以用正整数b_j来描述每个任务的难度。具体地，编号为i的人完成编号为j的任务所需要的时间为$a_i \times b_j$。

现在，你需要让n个人按编号依次完成m件任务，即第一个人从第一件任务开始，一直完成到最后一件；当第一个人完成第一件任务时，第二个人开始进行第一个任务……在此过程中，你需要保证一个任务只能正在由一个人完成。

你需要求出最后一个人完成最后一件任务的最早时间。

【数据规模和约定】

对于100%的数据，满足$1 \leq n,m \leq 100000$，$1 \leq a_i, b_j \leq 10000$。

【简要题解】

根据题意，我们可以设F_i表示第i个人的最早开始时间，其中$F_1 = 0$。另外，我们设$S_j = \sum_{k=1}^{j} b_k$。则我们容易得出以下DP式：

$$F_{i+1} - F_i = \max\{a_i \times S_{j+1} - a_{i+1} \times S_j\}$$

显然直接DP是过不去的，必须进行优化：

考虑构造向量$\vec{P} = (S_{j+1}, -S_j)$，向量$\vec{Q} = (a_i, a_{i+1})$。则我们只需要求向量$\vec{Q}$和所有的向量$\vec{P}$的点积最大值。显然这在凸壳上而且单峰，三分即可。