

# 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。

## 【输入文件】

输入文件为 *post.in*。

输入文件第一行为两个正整数 $n, m$ 。第二行 $n$ 个递增的整数，表示 $x_1 \sim x_n$ 。

## 【输出文件】

输出文件为 *post.out*。

输出一行一个整数，为最小的距离之和。

## 【输入输出样例】

post.in	post.out
10 5 1 2 3 6 7 9 11 22 44 50	9

## 【数据规模和约定】

对于30%的数据，满足 $n \leq 10$ 。

对于60%的数据，满足 $n \leq 60$ 。

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

# 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。

## 【输入文件】

输入文件为 `cs.in`。

本题含有多组数据，第一行为数据组数 $T$ 。

对于每组数据，第一行为一个正整数 $n$ ，第二行为 $n$ 个正整数，表示每个小兵的初始血量。

## 【输出文件】

输入文件为 `cs.out`。

对于每组数据，输出一行一个整数，表示你的最大补兵数。

## 【输入输出样例】

<code>cs.in</code>	<code>cs.out</code>
1 5 5 5 5 5 5	2

## 【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 $1 \leq n \leq 1000$ ， $1 \leq T \leq 70$ 。每个小兵的血量不超过1000。数据规模有一定的梯度。

# 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。

你要求出最后一个人完成最后一件任务的最早时间。

## 【输入文件】

输入文件为 *mission.in*。

第一行两个正整数 $m, n$ 。

第二行 $m$ 个正整数，分别表示 $b_1, b_2, \dots, b_m$ 。

第三行 $n$ 个正整数，分别表示 $a_1, a_2, \dots, a_n$ 。

## 【输出文件】

输入文件为 *mission.out*。

输出一行一个正整数，表示最后一个人完成最后一件任务的最早时间。

## 【输入输出样例】

<b>mission.in</b>	<b>mission.out</b>
3 4 1 2 3 3 4 1 2	40

## 【数据规模和约定】

对于40%的数据，满足 $n, m \leq 1000$ 。

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