

第 32 届全国信息学奥林匹克竞赛

CCF NOI 2015

第二试

竞赛时间：2015 年 7 月 19 日 8:00–13:00

题目名称	荷马史诗	品酒大会	小园丁与老司机
目录	epic	savour	farm
可执行文件名	epic	savour	farm
输入文件名	epic.in	savour.in	farm.in
输出文件名	epic.out	savour.out	farm.out
每个测试点时限	1 秒	1 秒	1 秒
内存限制	512MB	512MB	512MB
测试点数目	20	20	20
每个测试点分值	5	5	5
是否有部分分	是	否	是
题目类型	传统型	传统型	传统型
是否有附加文件	是	是	是

提交源程序须加后缀

对于 Pascal 语言	epic.pas	savour.pas	farm.pas
对于 C 语言	epic.c	savour.c	farm.c
对于 C++ 语言	epic.cpp	savour.cpp	farm.cpp

注意：最终测试时，所有编译命令均不打开任何优化开关。

荷马史诗

【问题描述】

追逐影子的人，自己就是影子 ——荷马

Allison 最近迷上了文学。她喜欢在一个慵懒的午后，细细地品上一杯卡布奇诺，静静地阅读她爱不释手的《荷马史诗》。但是由《奥德赛》和《伊利亚特》组成的鸿篇巨制《荷马史诗》实在是太长了，Allison 想通过一种编码方式使得它变得短一些。

一部《荷马史诗》中有 n 种不同的单词，从 1 到 n 进行编号。其中第 i 种单词出现的总次数为 w_i 。Allison 想要用 k 进制串 s_i 来替换第 i 种单词，使得其满足如下要求：

对于任意的 $1 \leq i, j \leq n$ ， $i \neq j$ ，都有： s_i 不是 s_j 的前缀。

现在 Allison 想要知道，如何选择 s_i ，才能使替换以后得到的新的《荷马史诗》长度最小。在确保总长度最小的情况下，Allison 还想知道最长的 s_i 的最短长度是多少？

一个字符串被称为 k 进制字符串，当且仅当它的每个字符是 0 到 $k-1$ 之间（包括 0 和 $k-1$ ）的整数。

字符串 $Str1$ 被称为字符串 $Str2$ 的前缀，当且仅当：存在 $1 \leq t \leq m$ ，使得 $Str1 = Str2[1..t]$ 。其中， m 是字符串 $Str2$ 的长度， $Str2[1..t]$ 表示 $Str2$ 的前 t 个字符组成的字符串。

【输入格式】

从文件 `epic.in` 中读入数据。

输入文件的第 1 行包含 2 个正整数 n, k ，中间用单个空格隔开，表示共有 n 种单词，需要使用 k 进制字符串进行替换。

接下来 n 行，第 $i+1$ 行包含 1 个非负整数 w_i ，表示第 i 种单词的出现次数。

【输出格式】

输出到文件 `epic.out` 中。

输出文件包括 2 行。

第 1 行输出 1 个整数，为《荷马史诗》经过重新编码以后的最短长度。

第 2 行输出 1 个整数，为保证最短总长度的情况下，最长字符串 s_i 的最短长度。

【样例输入 1】

```
4 2
1
1
2
```

2

【样例输出 1】

12

2

【样例说明 1】

用 $X_{(k)}$ 表示 X 是以 k 进制表示的字符串。

一种最优方案：令 $00_{(2)}$ 替换第 1 种单词， $01_{(2)}$ 替换第 2 种单词， $10_{(2)}$ 替换第 3 种单词， $11_{(2)}$ 替换第 4 种单词。在这种方案下，编码以后的最短长度为：

$$1 \times 2 + 1 \times 2 + 2 \times 2 + 2 \times 2 = 12$$

最长字符串 s_i 的长度为 2。

一种非最优方案：令 $000_{(2)}$ 替换第 1 种单词， $001_{(2)}$ 替换第 2 种单词， $01_{(2)}$ 替换第 3 种单词， $1_{(2)}$ 替换第 4 种单词。在这种方案下，编码以后的最短长度为：

$$1 \times 3 + 1 \times 3 + 2 \times 2 + 2 \times 1 = 12$$

最长字符串 s_i 的长度为 3。与最优方案相比，文章的长度相同，但是最长字符串的长度更长一些。

【样例输入 2】

6 3

1

1

3

3

9

9

【样例输出 2】

36

3

【样例说明 2】

一种最优方案：令 $000_{(3)}$ 替换第 1 种单词， $001_{(3)}$ 替换第 2 种单词， $01_{(3)}$ 替换第 3 种单词， $02_{(3)}$ 替换第 4 种单词， $1_{(3)}$ 替换第 5 种单词， $2_{(3)}$ 替换第 6 种单词。

【样例输入输出 3】

见选手目录下的 *epic/epic.in* 与 *epic/epic.ans*。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n 的规模	k 的规模	备注	约定
1	$n = 3$	$k = 2$		$0 < w_i \leq 10^{11}$
2	$n = 5$	$k = 2$		
3	$n = 16$	$k = 2$	所有 w_i 均相等	
4	$n = 1,000$	$k = 2$	w_i 在取值范围内均匀随机	
5	$n = 1,000$	$k = 2$		
6	$n = 100,000$	$k = 2$		
7	$n = 100,000$	$k = 2$	所有 w_i 均相等	
8	$n = 100,000$	$k = 2$		
9	$n = 7$	$k = 3$		
10	$n = 16$	$k = 3$	所有 w_i 均相等	
11	$n = 1,001$	$k = 3$	所有 w_i 均相等	
12	$n = 99,999$	$k = 4$	所有 w_i 均相等	
13	$n = 100,000$	$k = 4$		
14	$n = 100,000$	$k = 4$		
15	$n = 1,000$	$k = 5$		
16	$n = 100,000$	$k = 7$	w_i 在取值范围内均匀随机	
17	$n = 100,000$	$k = 7$		
18	$n = 100,000$	$k = 8$	w_i 在取值范围内均匀随机	
19	$n = 100,000$	$k = 9$		
20	$n = 100,000$	$k = 9$		

【提示】

选手请注意使用 64 位整数进行输入输出、存储和计算。

【评分方式】

对于每个测试点：

若输出文件的第 1 行正确，得到该测试点 40% 的分数；

若输出文件完全正确，得到该测试点 100% 的分数。

品酒大会

【问题描述】

一年一度的“幻影阁夏日品酒大会”隆重开幕了。大会包含品尝和趣味挑战两个环节，分别向优胜者颁发“首席品酒家”和“首席猎手”两个奖项，吸引了众多品酒师参加。

在大会的晚餐上，调酒师 Rainbow 调制了 n 杯鸡尾酒。这 n 杯鸡尾酒排成一行，其中第 i 杯酒 ($1 \leq i \leq n$) 被贴上了一个标签 s_i ，每个标签都是 26 个小写英文字母之一。设 $Str(l, r)$ 表示第 l 杯酒到第 r 杯酒的 $r - l + 1$ 个标签顺次连接构成的字符串。若 $Str(p, p_0) = Str(q, q_0)$ ，其中 $1 \leq p \leq p_0 \leq n, 1 \leq q \leq q_0 \leq n, p \neq q, p_0 - p + 1 = q_0 - q + 1 = r$ ，则称第 p 杯酒与第 q 杯酒是“ r 相似”的。当然两杯“ r 相似”($r > 1$) 的酒同时也是“1 相似”、“2 相似”、……、“ $(r - 1)$ 相似”的。特别地，对于任意的 $1 \leq p, q \leq n, p \neq q$ ，第 p 杯酒和第 q 杯酒都是“0 相似”的。

在品尝环节上，品酒师 Freda 轻松地评定了每一杯酒的美味度，凭借其专业的水准和经验成功夺取了“首席品酒家”的称号，其中第 i 杯酒 ($1 \leq i \leq n$) 的美味度为 a_i 。现在 Rainbow 公布了挑战环节的问题：本次大会调制的鸡尾酒有一个特点，如果把第 p 杯酒与第 q 杯酒调兑在一起，将得到一杯美味度为 $a_p a_q$ 的酒。现在请各位品酒师分别对于 $r = 0, 1, 2, \dots, n - 1$ ，统计出有多少种方法可以选出 2 杯“ r 相似”的酒，并回答选择 2 杯“ r 相似”的酒调兑可以得到的美味度的最大值。

【输入格式】

从文件 *savour.in* 中读入数据。

输入文件的第 1 行包含 1 个正整数 n ，表示鸡尾酒的杯数。

第 2 行包含一个长度为 n 的字符串 S ，其中第 i 个字符表示第 i 杯酒的标签。

第 3 行包含 n 个整数，相邻整数之间用单个空格隔开，其中第 i 个整数表示第 i 杯酒的美味度 a_i 。

【输出格式】

输出到文件 *savour.out* 中。

输出文件包括 n 行。第 i 行输出 2 个整数，中间用单个空格隔开。第 1 个整数表示选出两杯“ $(i - 1)$ 相似”的酒的方案数，第 2 个整数表示选出两杯“ $(i - 1)$ 相似”的酒调兑可以得到的最大美味度。若不存在两杯“ $(i - 1)$ 相似”的酒，这两个数均为 0。

【样例输入 1】

```
10
ponoiiipoi
```

2 1 4 7 4 8 3 6 4 7

【样例输出 1】

45 56
10 56
3 32
0 0
0 0
0 0
0 0
0 0
0 0
0 0
0 0

【样例说明 1】

用二元组 (p, q) 表示第 p 杯酒与第 q 杯酒。

0 相似：所有 45 对二元组都是 0 相似的，美味度最大的是 $8 \times 7 = 56$ 。

1 相似：(1,8) (2,4) (2,9) (4,9) (5,6) (5,7) (5,10) (6,7) (6,10) (7,10)，最大的 $8 \times 7 = 56$ 。

2 相似：(1,8) (4,9) (5,6)，最大的 $4 \times 8 = 32$ 。

没有 3,4,5, ..., 9 相似的两杯酒，故均输出 0。

【样例输入 2】

12
abaabaabaaba
1 -2 3 -4 5 -6 7 -8 9 -10 11 -12

【样例输出 2】

66 120
34 120
15 55
12 40
9 27
7 16
5 7
3 -4
2 -4
1 -4
0 0

0 0

【样例输入输出 3】

见选手目录下的 *savour/savour.in* 与 *savour/savour.ans*。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n 的规模	a_i 的规模	备注
1	$n = 100$	$ a_i \leq 10,000$	
2	$n = 200$		
3	$n = 500$		
4	$n = 750$		
5	$n = 1,000$	$ a_i \leq 1,000,000,000$	
6			
7	$n = 2,000$		
8			
9	$n = 99,991$	$ a_i \leq 1,000,000,000$	不存在“10 相似”的酒
10			
11	$n = 100,000$	$ a_i \leq 1000,000$	所有 a_i 的值都相等
12	$n = 200,000$		
13	$n = 300,000$		
14			
15	$n = 100,000$	$ a_i \leq 1,000,000,000$	
16			
17	$n = 200,000$		
18			
19	$n = 300,000$		
20			

小园丁与老司机

【问题描述】

小园丁 Mr. S 负责看管一片田野,田野可以看作一个二维平面。田野上有 n 棵许愿树,编号 $1,2,3,\dots,n$,每棵树可以看作平面上的一个点,其中第 i 棵树 ($1 \leq i \leq n$) 位于坐标 (x_i, y_i) 。任意两棵树的坐标均不相同。

老司机 Mr. P 从原点 $(0,0)$ 驾车出发,进行若干轮行动。每一轮,Mr. P 首先选择任意一个满足以下条件的方向:

1. 为左、右、上、左上 45° 、右上 45° 五个方向之一。
2. 沿此方向前进可以到达一棵他尚未许愿过的树。

完成选择后,Mr. P 沿该方向直线前进,必须到达该方向上距离最近的尚未许愿的树,在树下许愿并继续下一轮行动。如果没有满足条件的方向可供选择,则停止行动。他会采取最优策略,在尽可能多的树下许愿。若最优策略不唯一,可以选择任意一种。

不幸的是,小园丁 Mr. S 发现由于田野土质松软,老司机 Mr. P 的小汽车在每轮行进过程中,都会在田野上留下一条车辙印,一条车辙印可看作以两棵树(或原点和一棵树)为端点的一条线段。

在 Mr. P 之后,还有很多许愿者计划驾车来田野许愿,这些许愿者都会像 Mr. P 一样任选一种最优策略行动。Mr. S 认为非左右方向(即上、左上 45° 、右上 45° 三个方向)的车辙印很不美观,为了维护田野的形象,他打算租用一些轧路机,在这群许愿者到来之前夯实所有“可能留下非左右方向车辙印”的地面。

“可能留下非左右方向车辙印”的地面应当是田野上的若干条线段,其中每条线段都包含在某一种最优策略的行进路线中。每台轧路机都采取满足以下三个条件的工作模式:

1. 从原点或任意一棵树出发。
2. 只能向上、左上 45° 、右上 45° 三个方向之一移动,并且只能在树下改变方向或停止。
3. 只能经过“可能留下非左右方向车辙印”的地面,但是同一块地面可以被多台轧路机经过。

现在 Mr. P 和 Mr. S 分别向你提出了一个问题:

1. 请给 Mr. P 指出任意一条最优路线。
2. 请告诉 Mr. S 最少需要租用多少台轧路机。

【输入格式】

从文件 *farm.in* 中读入数据。

输入文件的第 1 行包含 1 个正整数 n ,表示许愿树的数量。

接下来 n 行,第 $i+1$ 行包含 2 个整数 x_i, y_i ,中间用单个空格隔开,表示第 i 棵许愿树的坐标。

【输出格式】

输出到文件 *farm.out* 中。

输出文件包括 3 行。

输出文件的第 1 行输出 1 个整数 m ，表示 Mr. P 最多能在多少棵树下许愿。

输出文件的第 2 行输出 m 个整数，相邻整数之间用单个空格隔开，表示 Mr. P 应该依次在哪些树下许愿。

输出文件的第 3 行输出 1 个整数，表示 Mr. S 最少需要租用多少台轧路机。

【样例输入 1】

```
6
-1 1
1 1
-2 2
0 8
0 9
0 10
```

【样例输出 1】

```
3
2 1 3
3
```

【样例说明 1】

最优路线 2 条可许愿 3 次： $(0,0) \rightarrow (1,1) \rightarrow (-1,1) \rightarrow (-2,2)$ 或 $(0,0) \rightarrow (0,8) \rightarrow (0,9) \rightarrow (0,10)$ 。

至少 3 台轧路机，路线是 $(0,0) \rightarrow (1,1)$ ， $(-1,1) \rightarrow (-2,2)$ 和 $(0,0) \rightarrow (0,8) \rightarrow (0,9) \rightarrow (0,10)$ 。

【样例输入 2】

```
4
0 1
-2 1
2 1
3 2
```

【样例输出 2】

```
4
1 2 3 4
2
```

【样例说明 2】

最优路线唯一： $(0,0) \rightarrow (0,1) \rightarrow (-2,1) \rightarrow (2,1) \rightarrow (3,2)$ ，可许愿 4 次。其中在 $(0,1)$ 许愿后，从 $(-2,1)$ 出发沿着向右的方向能够到达的最近的未许愿过的树是 $(2,1)$ ，所以可以到达 $(2,1)$ 。

而如果沿着 $(0,0) \rightarrow (0,1) \rightarrow (2,1) \rightarrow (-2,1)$ 的方向前进，此时 $(-2,1)$ 右边所有树都是许愿过的，根据题目条件规定，停止前进。故无法获得最优解。

$(0,0) \rightarrow (0,1)$ 与 $(2,1) \rightarrow (3,2)$ 会留下非左右方向车辙印，需 2 台轧路机。

【样例输入输出 3】

见选手目录下的 *farm/farm.in* 与 *farm/farm.ans*。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	n 的规模	x_i, y_i 的规模	备注
1	$n = 5$	$ x_i \leq 100$ $0 < y_i \leq 100$	
2	$n = 10$		
3	$n = 100$	$ x_i \leq 10,000$ $0 < y_i \leq 10,000$	
4	$n = 1,000$		
5	$n = 5,000$	$ x_i \leq 1,000,000$ $0 < y_i \leq 1,000,000$	保证最优路线唯一
6			
7			
8	$n = 5,000$	$ x_i \leq 1,000,000$ $0 < y_i \leq 1,000,000$	保证 y_i 互不相同
9	$n = 50,000$		
10			
11	$n = 5,000$	$ x_i \leq 1,000,000$ $0 < y_i \leq 1,000,000$	保证对于任意整数 Y ， 满足 $y_i = Y$ 的树不超过 1,000 棵 存在一种最优解，使得 轧路机不重复经过同一路面
12			
13	$n = 50,000$		
14			
15	$n = 10,000$	$ x_i \leq 1,000,000,000$ $0 < y_i \leq 1,000,000,000$	保证对于任意整数 Y ， 满足 $y_i = Y$ 的树不超过 1,000 棵
16			
17	$n = 30,000$		
18			
19	$n = 50,000$		
20			

【评分方式】

对于每个测试点：

若输出文件的第 1 行正确，得到该测试点 20% 的分数；
若输出文件的前两行正确，得到该测试点 40% 的分数；
若输出文件完全正确，得到该测试点 100% 的分数。