

# USACO 试题精选

马融

2014 年 7 月 21 日

# Contents

<b>1</b>	<b>贪心</b>	<b>7</b>
1	★☆☆☆☆ 酸奶工厂 . . . . .	8
2	★☆☆☆☆ 购买饲料 . . . . .	11
3	★☆☆☆☆ 公司利润 . . . . .	14
4	★★☆☆☆ 奶牛工资 . . . . .	19
<b>2</b>	<b>分治法</b>	<b>22</b>
5	★☆☆☆☆ 分期付款 . . . . .	23
6	★★☆☆☆ 享受巧克力 . . . . .	26
7	★★☆☆☆ 愤怒的奶牛 . . . . .	29
8	★★☆☆☆ 跳石头 . . . . .	31
9	★★★☆☆ 破坏阴谋 . . . . .	35
<b>3</b>	<b>堆</b>	<b>38</b>
10	★★☆☆☆ 锯栅栏 . . . . .	39
11	★★★☆☆ 奶牛探险 . . . . .	43
12	★★★☆☆ 赞助学费 . . . . .	45
13	★★★★☆ 奶牛优惠券 . . . . .	47
14	★★★★☆ 都市地平线 . . . . .	49
<b>4</b>	<b>数值计算</b>	<b>51</b>
15	★☆☆☆☆ 负二进制 . . . . .	52
16	★★☆☆☆ 分数变小数 . . . . .	55
17	★★☆☆☆ 数的幂次 . . . . .	58
<b>5</b>	<b>排序</b>	<b>60</b>
18	★★☆☆☆ 重排干草 . . . . .	61
<b>6</b>	<b>排序 (二)</b>	<b>63</b>
19	★☆☆☆☆ 自私的奶牛 . . . . .	64
20	★★☆☆☆ 录制节目 . . . . .	66
21	★★★☆☆ 雄伟的山峦 . . . . .	68

<b>7 排序（三）</b>	<b>70</b>
22 ★☆☆☆☆ 分数排序 . . . . .	71
23 ★★★★★ 奶牛集体照 . . . . .	74
<b>8 归并排序</b>	<b>76</b>
24 ★★★★★ 整理绳子 . . . . .	77
25 ★★★★★ 奶牛集会 . . . . .	79
<b>9 贪心——进阶</b>	<b>81</b>
26 ★☆☆☆☆ 保护花朵 . . . . .	82
27 ★★★★★ 奶牛杂技 . . . . .	84
28 ★★★★★ 二道工序 . . . . .	86
<b>10 栈</b>	<b>88</b>
29 ★☆☆☆☆ 括号序列 . . . . .	89
30 ★☆☆☆☆ 向右看齐 . . . . .	91
<b>11 队列</b>	<b>93</b>
31 ★☆☆☆☆ 失误的滑雪 . . . . .	94
32 ★★★★★ 丑数 . . . . .	96
33 ★☆☆☆☆ 宴会邀请 . . . . .	99
<b>12 穷举</b>	<b>101</b>
34 ★☆☆☆☆ 翻转奶牛 . . . . .	102
35 ★☆☆☆☆ 翻转棋 . . . . .	104
<b>13 数位问题</b>	<b>106</b>
36 ★★★★★ 数数的梦 . . . . .	107
37 ★★★★★ 二进制编号 . . . . .	109
<b>14 深度优先搜索</b>	<b>111</b>
38 ★☆☆☆☆ 象棋皇后 . . . . .	112
39 ★★★★★ 拼图游戏 . . . . .	114
40 ★★★★★ 埃及分数 . . . . .	117
41 ★★★★★ 栅栏木料 . . . . .	120
42 ★★★★★ 解密牛语 . . . . .	122
<b>15 TRIE</b>	<b>124</b>
43 ★★★★★ 时间旅行 . . . . .	125
44 ★★★★★ 一号单词 . . . . .	128

<b>16 并查集</b>	<b>130</b>
45   ★★★★☆☆ 真假奶牛 . . . . .	131
46   ★★★★☆☆ 工作安排 . . . . .	133
47   ★★★★☆☆ 导航难题 . . . . .	135
48   ★★★★☆☆ 叠积木 . . . . .	138
<b>17 哈希</b>	<b>141</b>
49   ★★★★☆☆ 电灯 . . . . .	142
50   ★★★★★★ 宝藏图 . . . . .	144
<b>18 树状数组</b>	<b>146</b>
51   ★★★★☆☆ 奶牛抗议 . . . . .	147
52   ★★★★☆☆ 奶牛赛跑 . . . . .	149
53   ★★★★★★ 道路植草 . . . . .	151
<b>19 线段树</b>	<b>153</b>
54   ★★★★☆☆ 奶牛猜数 . . . . .	154
55   ★★★★☆☆ 跨栏训练 . . . . .	156
56   ★★★★☆☆ 预定旅馆 . . . . .	158
57   ★★★★★★ 修路 . . . . .	161
<b>20 平衡树</b>	<b>163</b>
58   ★★★★☆☆ 挑剔的美食家 . . . . .	164
59   ★★★★☆☆ 集市班车 . . . . .	166
60   ★★★★★★ 奶牛登机（缺） . . . . .	168
61   ★★★★★★ 期末总评 . . . . .	170
<b>21 后缀数组</b>	<b>172</b>
62   ★★★★☆☆ 最佳队名 . . . . .	173
<b>22 Floodfill</b>	<b>175</b>
63   ★★☆☆☆☆ 池塘计数 . . . . .	176
64   ★★☆☆☆☆ 卫星图片 . . . . .	178
65   ★★☆☆☆☆ 奶牛万圣节 . . . . .	180
<b>23 最小生成树</b>	<b>182</b>
66   ★★☆☆☆☆ 建造道路 . . . . .	183
67   ★★★★☆☆ 奶牛施工队 . . . . .	185
<b>24 宽度优先遍历</b>	<b>187</b>
68   ★★☆☆☆☆ 拖拉机 . . . . .	188
69   ★★★★☆☆ 玉米迷宫 . . . . .	190
70   ★★★★☆☆ 奶牛攀岩 . . . . .	192

71	★★★★★	黄金莲花池 . . . . .	194
<b>25 Bellman-Ford</b>			<b>196</b>
72	★★☆☆☆	虫洞 . . . . .	197
73	★★★★☆☆	奶牛观光 . . . . .	199
74	★★★★☆☆	奶牛的站位 . . . . .	201
<b>26 Dijkstra</b>			<b>203</b>
75	★★★★☆☆	领取巧克力 . . . . .	204
76	★★★★☆☆	次短路径 . . . . .	206
77	★★★★☆☆	道路翻新 . . . . .	208
78	★★★★★	公路和航线 . . . . .	210
<b>27 Floyd</b>			<b>212</b>
79	★★☆☆☆	奶牛比赛 . . . . .	213
80	★★☆☆☆	奶牛跨栏 . . . . .	215
81	★★★★☆	过路费 . . . . .	217
82	★★★★☆	奶牛接力 . . . . .	219
<b>28 深度优先遍历</b>			<b>221</b>
83	★★★★☆	明星奶牛 . . . . .	222
84	★★★★☆	冗余路径 . . . . .	224
<b>29 匹配</b>			<b>226</b>
85	★★★★☆☆	小行星群 . . . . .	227
86	★★★★☆	泥泞的牧场 . . . . .	229
<b>30 网络流</b>			<b>231</b>
87	★★★★☆☆	奶牛晚宴 . . . . .	232
88	★★★★☆	游览牧场 . . . . .	234
89	★★★★☆	地震破坏 . . . . .	236
90	★★★★★	玩具消毒 . . . . .	238
<b>31 动态规划（一）</b>			<b>240</b>
91	★★☆☆☆	接住苹果 . . . . .	241
92	★★☆☆☆	方形牛棚 . . . . .	243
93	★★★★☆☆	滑雪课程 . . . . .	245
94	★★★★☆☆	滑雪比赛 . . . . .	247
<b>32 动态规划（二）</b>			<b>249</b>
95	★★☆☆☆	奶牛飞盘队 . . . . .	250
96	★★★★☆☆	滑雪缆车 . . . . .	252
97	★★★★☆☆	奶牛自行车队 . . . . .	254

98	★★★★☆	打扫食槽 . . . . .	256
99	★★★★☆	连击游戏 . . . . .	258
<b>33</b>		<b>动态规划（三）——区间型</b>	<b>260</b>
100	★★☆☆☆	金币游戏 . . . . .	261
101	★★★★☆	修改回文 . . . . .	264
102	★★★★☆	奶牛吃草 . . . . .	266
103	★★★★☆	提交作业 . . . . .	268
<b>34</b>		<b>动态规划（四）——背包问题</b>	<b>270</b>
104	★★☆☆☆	平分子集 . . . . .	271
105	★★★★☆	太空电梯 . . . . .	273
106	★★★★☆	奶牛会展 . . . . .	275
107	★★★★☆	促销日 . . . . .	277
108	★★★★☆	股票市场 . . . . .	279
<b>35</b>		<b>动态规划（五）——背包问题续</b>	<b>281</b>
109	★★★★☆	三角牧场 . . . . .	282
110	★★★★☆	录制唱片 . . . . .	284
111	★★★★☆	电子游戏 . . . . .	286
112	★★★★☆	三个代表 . . . . .	288
113	★★★★★	最少找零 . . . . .	290
<b>36</b>		<b>动态规划（六）——树型 DP</b>	<b>292</b>
114	★★★★☆	圣诞装饰 . . . . .	293
115	★★★★☆	拜访奶牛 . . . . .	295
116	★★★★☆	手机网络 . . . . .	297
117	★★★★☆	切断道路 . . . . .	299
118	★★★★★	产奶比赛 . . . . .	301
119	★★★★★	焊接 . . . . .	303
<b>37</b>		<b>动态规划（七）——单调队列优化</b>	<b>305</b>
120	★★★★☆	修剪草坪 . . . . .	306
121	★★★★☆	玉米实验 . . . . .	308
122	★★★★☆	干草金字塔 . . . . .	310
123	★★★★☆	又买饲料 . . . . .	312
124	★★★★★	土地并购 . . . . .	314
125	★★★★★	约翰的书架 . . . . .	316
<b>38</b>		<b>动态规划（八）——位向量型</b>	<b>318</b>
126	★★★★☆	混乱奶牛 . . . . .	319
127	★★★★☆	全能奶牛 . . . . .	321

128	★★★★☆	洞穴奶牛 . . . . .	323
129	★★★☆☆	不找零 . . . . .	325
130	★★★★☆	奶牛乘电梯 . . . . .	327
131	★★★★☆	安排牧场 . . . . .	329

# Chapter 1

## 贪心

### 知识点

- 贪心思想
- 简易的递推公式
- 选择排序



## 1 ★☆☆☆☆ 酸奶工厂

奶牛经营了一家酸奶工厂，生意非常红火。在接下去的  $N$  个月里，第  $i$  个月需要向社会提供  $A_i$  吨酸奶。酸奶的生产受到很多因素的影响，所以每个月的生产成本是变化的，其中第  $i$  个月的成本是每吨  $C_i$  元。

奶牛可以提前里把酸奶做好，存在仓库里，等需要的时候再拿出来卖。存储在仓库里的酸奶，每吨酸奶存放一个月需要支付  $S$  元的维护费用，存放的时间可以任意长。假设工厂的产量是无限的，存储酸奶的仓库也是无限大的。请问为了满足订单的需要，奶牛生产这些酸奶最少要花多少钱？

### 输入格式

- 第一行：两个整数  $N$  和  $S$ ， $1 \leq N \leq 10000$ ,  $1 \leq S \leq 100$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $C_i$  和  $A_i$ ， $1 \leq C_i \leq 5000$ ,  $1 \leq A_i \leq 10000$

### 输出格式

- 单个整数：表示生产酸奶的最小总费用

### 样例输入

```
4 5
88 200
89 400
97 300
91 500
```

### 样例输出

```
126900
```

### 解释

第一个月生产 200 吨酸奶；第二个月生产 700 吨酸奶，并存下 300 吨；第三个月不生产酸奶；第三个月生产 500 吨

## 来源

Yogurt factory, 2005 Mar

## 代号

yogfac

## 解答

---

**Algorithm 1:** yogfac

---

```
1  $ans \leftarrow 0$ 
2  $opt \leftarrow \infty$ 
3 for  $i \leftarrow 1$  to  $N$  do
4   if  $opt > C[i]$  then
5      $opt \leftarrow C[i]$ 
6    $ans \leftarrow ans + opt \cdot A[i]$ 
7    $opt \leftarrow opt + opt \cdot S$ 
8 end for
9 print  $ans$ 
```

---

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int inf = 5000;
4  int main() {
5      int n, s;
6      cin >> n >> s;
7      long long ans = 0;
8      long long opt = inf;
9      for (int i = 0; i < n; i++) {
10         int c, a;
11         cin >> c >> a;
12         if (opt > c) {
13             opt = c;
14         }
15         ans += opt * a;
16         opt += s;
17     }
18     cout << ans << endl;
19     return 0;
20 }
```

```
1  const
2      inf = 5000;
3  var
4      n, s, i, a, c : longint;
5      ans, opt : int64;
6  begin
7      read(n, s);
8      opt := inf;
9      for i := 1 to n do begin
10         read(c, a);
11         if opt > c then
12             opt := c;
13         inc(ans, opt * a);
14         inc(opt, s);
15     end;
16     writeln(ans);
17 end.
```

## 教学要点

1. （抛砖引玉）给出一个平凡算法（比如  $O(N^2)$  的），目的在于帮助学生理解题意
2. 用具体例子（比如样例），使学生发现改进到算法的思路
3. 引导学生讨论，由老师形成完整的结论，给出伪代码
4. （处理细节）指导学生编程，注意  $\infty$  和 64 位整数
5. 过程在 1 小时左右

## 2 ★☆☆☆☆ 购买饲料

约翰开车回家，遇到了双十一节，那么就顺路买点饲料吧。回家的路程一共有  $E$  公里，这一路上会经过  $N$  家商店，第  $i$  家店里有  $F_i$  吨饲料，售价为每吨  $C_i$  元。约翰打算买  $K$  吨饲料，他知道商家的库存是足够的，至少所有店的库存总和不会少于  $K$ 。除了购买饲料要钱，运送饲料也是要花油钱的，约翰的卡车上如果装着  $X$  吨饲料，那么他行驶一公里会花掉  $X$  元，行驶  $D$  公里需要  $DX$  元。已知第  $i$  家店距约翰所在的起点有  $X_i$  公里，那么约翰在哪些商店买饲料运回家，才能做到最省钱呢？

### 输入格式

- 第一行：三个整数  $K$ ， $E$  和  $N$ ， $1 \leq K \leq 100, 1 \leq E \leq 350, 1 \leq N \leq 100$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有三个整数  $X_i$ ， $F_i$  和  $C_i$ ， $0 < X_i < E, 1 \leq F_i \leq 100, 1 \leq C_i \leq 10^6$

### 输出格式

- 单个整数：表示购买和运送饲料的最小费用之和

### 样例输入

```
2 5 3
3 1 2
4 1 2
1 1 1
```

### 样例输出

```
7
```

### 解释

在离家较近的两家商店里各购买一吨饲料，则花在路上的钱是  $1 + 2 = 3$ ，花在店里的钱是  $2 + 2 = 4$

## 来源

Buying Feed II, 2010 Jan

## 代号

feed

## 解答

---

**Algorithm 2:** feed

---

1	<b>for</b> $i \leftarrow 1$ <b>to</b> $N$ <b>do</b>	12	$ans \leftarrow 0$
2	$P[i] \leftarrow C[i] + (E - X[i])$	13	<b>for</b> $i \leftarrow 1$ <b>to</b> $N$ <b>do</b>
3	<b>end for</b>	14	<b>if</b> $K > F[i]$ <b>then</b>
4	<b>for</b> $i \leftarrow 1$ <b>to</b> $N$ <b>do</b>	15	$ans \leftarrow ans + P[i] \cdot F[i]$
5	<b>for</b> $j \leftarrow i + 1$ <b>to</b> $N$ <b>do</b>	16	$K \leftarrow K - F[i]$
6	<b>if</b> $P[i] > P[j]$ <b>then</b>	17	<b>else</b>
7	$P[i] \leftrightarrow P[j]$	18	$ans \leftarrow ans + P[i] \cdot K$
8	$F[i] \leftrightarrow F[j]$	19	<b>break</b>
9	<b>end if</b>	20	<b>end if</b>
10	<b>end for</b>	21	<b>end for</b>
11	<b>end for</b>	22	<b>print</b> $ans$

---

## 教学要点

- 利用网络购物的经验，引导学生找到将运费算入购物成本的方法
- 容易想到贪心法，但需要引导学生区分库存不足和库存充足时的做法
- 得到算法框架后，注意检查学生的程序逻辑，循环控制与  $K$  和  $i$  都有关，可选任意一个作为循环变量，但另一个如何处理是关键。
- 实现时，注意学生是否在排序时遗忘了交换  $F$ 。
- 整个过程在 1 小时左右

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 100;
4  int f[maxn], p[maxn];
5  int main() {
6      int k, e, n;
7      cin >> k >> e >> n;
8      for (int i = 0; i < n; i++) {
9          int x, c;
10         cin >> x >> f[i] >> c;
11         p[i] = c + (e - x);
12     }
13     for (int i = 0; i < n; i++) {
14         for (int j = i + 1; j < n; j++) {
15             if (p[i] > p[j]) {
16                 int t = p[i]; p[i] = p[j]; p[j] = t;
17                 t = f[i]; f[i] = f[j]; f[j] = t;
18             }
19         }
20     }
21     int ans = 0;
22     for (int i = 0; k > 0; i++) {
23         if (k > f[i]) {
24             ans += p[i] * f[i];
25             k -= f[i];
26         } else {
27             ans += p[i] * k;
28             k = 0;
29         }
30     }
31     cout << ans << endl;
32     return 0;
33 }
```

```
1  const
2      maxn = 100;
3  var
4      n, k, e, i, j, x, c, ans : longint;
5      f, p : array[1..maxn] of longint;
6  begin
7      read(k, e, n);
8      for i := 1 to n do begin
9          read(x, f[i], c);
10         p[i] := c + (e - x);
11     end;
12     for i := 1 to n do
13         for j := i + 1 to n do
14             if p[i] > p[j] then begin
15                 x := p[i]; p[i] := p[j]; p[j] := x;
16                 x := f[i]; f[i] := f[j]; f[j] := x;
17             end;
18     for i := 1 to n do
19         if k > f[i] then begin
20             inc(ans, p[i] * f[i]);
21             dec(k, f[i]);
22         end else begin
23             inc(ans, p[i] * k);
24             break;
25         end;
26     writeln(ans);
27 end.
```

### 3 ★☆☆☆☆ 公司利润

奶牛开了家公司，已经连续运作了  $N$  天。它们在第  $i$  天获得了  $A_i$  元的利润，不过有些天是亏钱的，这种情况下利润就是一个负数。约翰想为它们写个新闻，吹嘘它们的惊人业绩。请你帮助他选出一段连续的日子，长度不限，但至少要是包括一天，使得公司在这段日子中的利润之和最大。

#### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $-1000 \leq A_i \leq 1000$

#### 输出格式

- 单个整数：表示最大的利润之和

#### 样例输入

7  
-3  
4  
9  
-2  
-5  
8  
-3

#### 样例输出

14

#### 解释

选择第三天到第六天， $4 + 9 - 2 - 5 + 8 = 14$

## 来源

Profits, 2011 Jan

## 代号

profits

## $O(N^3)$ 的算法

每个子序列有自己的起点和终点，所以枚举子序列的起点和终点就可以枚举出所有的连续的子序列。逐个计算它们的和，求出其中最大的一个。所要计算的内容可以表示为：

$$ans = \max_{1 \leq i \leq j \leq N} \{A_i + A_{i+1} + \cdots + A_j\}$$

---

**Algorithm 3:** profits —  $O(N^3)$ 

---

```
1  $ans \leftarrow -\infty$ 
2 for  $i \leftarrow 1$  to  $N$  do
3   for  $j \leftarrow i$  to  $N$  do
4      $sum \leftarrow 0$ 
5     for  $k \leftarrow i$  to  $j$  do
6        $sum \leftarrow sum + A[k]$ 
7     end for
8     if  $ans < sum$  then
9        $ans \leftarrow sum$ 
10  end for
11 end for
12 print  $ans$ 
```

---

**思考题 1** 写程序时如何解决  $-\infty$ ？

**思考题 2** 为何这个算法的时间复杂度是  $O(N^3)$  的？能否比较精确地求出算法执行了多少次运算？

## $O(N^2)$ 的算法

“部分和”（Partial Sum）可以更快地计算一段连续序列的和。所谓“部分和”就是一个新的序列  $\{S_i\}$ ， $S_1 = A_1$ ， $S_2 = A_1 + A_2$ ， $S_3 = A_1 + A_2 + A_3$ ， $S_i$  记录了  $\{A_i\}$  中前  $i$  项的和。 $\{A_i\}$  的求和问题可以转化为  $\{S_i\}$  的求差问题，比如

$$A_3 + A_4 = (A_1 + A_2 + A_3 + A_4) - (A_1 + A_2) = S_4 - S_2$$



一般性的计算公式是

$$A_i + A_{i+1} + \cdots + A_{j-1} + A_j = S_j - S_{i-1}$$

所要计算的内容可以表示为:

$$ans = \max_{1 \leq i \leq j \leq N} \{S_j - S_{i-1}\}$$

---

**Algorithm 4:** profits —  $O(N^2)$

---

```
1  $ans \leftarrow -\infty$ 
2  $S[0] \leftarrow 0$ 
3 for  $i \leftarrow 1$  to  $N$  do
4    $S[i] \leftarrow S[i-1] + A[i]$ 
5 end for
6 for  $i \leftarrow 1$  to  $N$  do
7   for  $j \leftarrow i+1$  to  $N$  do
8     if  $ans < S[j] - S[i-1]$  then
9        $ans \leftarrow S[j] - S[i-1]$ 
10  end for
11 end for
12 print  $ans$ 
```

---

**思考题 3** 为何需要设置  $S[0] = 0$ ?

$O(N)$  的算法

---

**Algorithm 5:** profits —  $O(N)$

---

```
1  $ans \leftarrow -\infty$ 
2  $last \leftarrow 0$ 
3 for  $i \leftarrow 1$  to  $N$  do
4   if  $last < 0$  then
5      $last \leftarrow 0$ 
6    $last \leftarrow last + A[i]$ 
7   if  $ans < last$  then
8      $ans \leftarrow last$ 
9 end for
10 print  $ans$ 
```

---

## 评注

这是一个经典问题，名为最大连续子段和（Maximum Subarray Problem）。相关题目有

- 2009 年 NOIP 提高组和普及组的初赛程序填空
- 2013 年 NOIP 普及组复赛第三题——“小朋友的数字”

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int inf = 1000 * 10000;
4  int main() {
5      int n;
6      cin >> n;
7      int last = 0;
8      int ans = -inf;
9      for (int i = 0; i < n; i++) {
10         int a;
11         cin >> a;
12         if (last < 0) {
13             last = 0;
14         }
15         last += a;
16         if (ans < last) {
17             ans = last;
18         }
19     }
20     cout << ans << endl;
21     return 0;
22 }
```

```
1  const
2      maxn = 100000;
3      inf = 1000 * maxn;
4  var
5      n, i, ans, last, a : longint;
6  begin
7      read(n);
8      ans := -inf;
9      for i := 1 to n do begin
10         read(a);
11         if last < 0 then
12             last := 0;
13         inc(last, a);
14         if ans < last then
15             ans := last;
16         end;
17         writeln(ans);
18     end.
```

## 4 ★★☆☆☆☆ 奶牛工资

贝西工作勤勤恳恳，她每月向约翰索要  $C$  元钱作为工资。约翰手上有不少钱，他一共有  $N$  种面额的钞票。第  $i$  种钞票的面额记作  $V_i$ ，约翰有  $K_i$  张。钞票的面额设定是比较合理的，保证所有大面额的钞票都是所有小面额钞票的整数倍。假设约翰每个月给贝西发一次工资，那么这些钱够发几个月的工资呢？贝西不会找零，如果约翰发的钱大于  $C$  元，多余的部分就算是贝西的奖励了。

### 输入格式

- 第一行：两个整数  $N$  和  $C$ ， $1 \leq N \leq 20, 1 \leq C \leq 10^9$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $V_i$  和  $K_i$ ， $1 \leq V_i \leq 10^9, 1 \leq K_i \leq 10^6$

### 输出格式

- 单个整数：表示约翰最多能给贝西发几个月的工资

### 样例输入

```
3 6
10 1
1 100
5 120
```

### 样例输出

```
111
```

### 解释

第一个月先给一张十元的，接下来十个月每个月都给两张五元的，最后一百个月每月给一张一元的和一张五元的

## 来源

Allowance, 2005 Oct

## 代号

allow

## 解答

Function test	
1 $pay \leftarrow C$	12 <b>if</b> $pay \leq 0$ <b>then</b>
2 <b>for</b> $i \leftarrow N$ <b>downto</b> 1 <b>do</b>	13 <b>return</b> true
3 $t \leftarrow \lfloor pay/V[i] \rfloor$	14 <b>for</b> $i \leftarrow 1$ <b>to</b> $N$ <b>do</b>
4 <b>if</b> $t > K[i]$ <b>then</b>	15 <b>if</b> $K[i] > 1$ <b>then</b>
5 $pay \leftarrow pay - V[i] \cdot K[i]$	16 $K[i] \leftarrow K[i] - 1$
6 $K[i] \leftarrow 0$	17 <b>return</b> true
7 <b>else</b>	18 <b>end if</b>
8 $pay \leftarrow pay - V[i] \cdot t$	19 <b>end for</b>
9 $K[i] \leftarrow K[i] - t$	20 <b>return</b> false
10 <b>end if</b>	
11 <b>end for</b>	

  

Algorithm 6: allow	
1 <b>for</b> $i \leftarrow 1$ <b>to</b> $N$ <b>do</b>	9 $ans \leftarrow 0$
2 <b>for</b> $j \leftarrow i + 1$ <b>to</b> $N$ <b>do</b>	10 <b>while</b> test() <b>do</b>
3 <b>if</b> $V[i] > V[j]$ <b>then</b>	11 $ans \leftarrow ans + 1$
4 $V[i] \leftrightarrow V[j]$	12 <b>end while</b>
5 $K[i] \leftrightarrow K[j]$	13 <b>return</b> $ans$
6 <b>end if</b>	
7 <b>end for</b>	
8 <b>end for</b>	

## 教学要点

- test 写成函数的方法可以让结构更清晰，不然在循环到一半的过程中用 if 和 break 控制退出比较难看

## 参考程序

```

1  #include <iostream>
2  using namespace std;
3  const int maxn = 20;
4  int v[maxn], k[maxn];
5  int n, c;
6
7  bool test(void) {
8      int pay = c;
9      for (int i = n - 1; i >= 0; i--) {
10         int t = pay / v[i];
11         if (t > k[i]) {
12             pay -= v[i] * k[i];
13             k[i] = 0;
14         } else {
15             pay -= v[i] * t;
16             k[i] -= t;
17         }
18     }
19     if (pay <= 0) return true;
20     for (int i = 0; i < n; i++) {
21         if (k[i] > 0) {
22             k[i]--;
23             return true;
24         }
25     }
26     return false;
27 }
28
29 int main () {
30     cin >> n >> c;
31     for (int i = 0; i < n; i++) {
32         cin >> v[i] >> k[i];
33     }
34     for (int i = 0; i < n; i++) {
35         for (int j = i + 1; j < n; j++) {
36             if (v[i] > v[j]) {
37                 int t = v[i]; v[i] = v[j]; v[j] = t;
38                 t = k[i]; k[i] = k[j]; k[j] = t;
39             }
40         }
41     }
42     int ans = 0;
43     while (test()) {
44         ans++;
45     }
46     cout << ans << endl;
47     return 0;
48 }

```

```

1  const
2      maxn = 20;
3  var
4      n, c, i, j, ans, t : longint;
5      v, k : array[1..maxn] of longint;
6
7  function test:boolean;
8  var
9      pay, i, t : longint;
10 begin
11     pay := c;
12     for i := n downto 1 do begin
13         t := pay div v[i];
14         if t > k[i] then begin
15             dec(pay, v[i] * k[i]);
16             k[i] := 0;
17         end else begin
18             dec(pay, v[i] * t);
19             dec(k[i], t);
20         end;
21     end;
22     if pay <= 0 then exit(true);
23     for i := 1 to n do
24         if k[i] > 0 then begin
25             dec(k[i]);
26             exit(true);
27         end;
28     exit(false);
29 end;
30
31 begin
32     read(n, c);
33     for i := 1 to n do
34         read(v[i], k[i]);
35     for i := 1 to n do
36         for j := i + 1 to n do
37             if v[i] > v[j] then begin
38                 t := v[i]; v[i] := v[j]; v[j] := t;
39                 t := k[i]; k[i] := k[j]; k[j] := t;
40             end;
41         while test do
42             inc(ans);
43             writeln(ans);
44     end.

```

## Chapter 2

# 分治法

## 5 ★☆☆☆☆ 分期付款

约翰刷了  $N$  次信用卡，第  $i$  次刷掉了  $A_i$  元。银行告诉他可以在接下来的  $M$  个月里分期还清这些债务，但是先发生的债务必须先偿还，也不能把一笔债务分在两个月里还。约翰想找到一种方法，使得每个月还的钱尽量均匀。请问，约翰应该在每个月还多少债务，才能使他在所有月份中的最大还款数额最小？

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq M \leq N \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $1 \leq A_i \leq 10000$

### 输出格式

- 单个整数：表示最大还款额的最小值

### 样例输入

7 5  
100  
400  
300  
100  
500  
101  
400

### 样例输出

500

### 解释

前两笔放在第一个月，第三和第四笔放在第二个月，最后三笔各自放在一个月



## 来源

Monthly Expense, 2007 Mar

## 代号

expense

## 解答

二分法。记所求答案为  $ans$ ，设  $ans \in (low, high]$ 。

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 100000;
4  const int inf = 10000 * maxn;
5  int n, m;
6  int a[maxn];
7
8  bool test(int x) {
9      int i = 0;
10     for (int k = 0; k < m; k++) {
11         int s = 0;
12         while (i < n && s + a[i] <= x) {
13             s += a[i];
14             i++;
15         }
16     }
17     return (i >= n);
18 }
19
20 int main() {
21     cin >> n >> m;
22     for (int i = 0; i < n; i++) {
23         cin >> a[i];
24     }
25     int low = 0;
26     int high = inf;
27     while (high - low > 1) {
28         int mid = (high + low) / 2;
29         if (test(mid)) {
30             high = mid;
31         } else {
32             low = mid;
33         }
34     }
35     cout << high << endl;
36     return 0;
37 }
```

```
1  const
2      maxn = 100000;
3      inf = maxn * 10000;
4  var
5      n, m, i, low, high, mid : longint;
6      a : array[1..maxn] of longint;
7
8  function test(x:longint):boolean;
9  var
10     i, k, s : longint;
11 begin
12     i := 1;
13     for k := 1 to m do begin
14         s := 0;
15         while (i <= n) and (s + a[i] <= x) do begin
16             inc(s, a[i]);
17             inc(i);
18         end;
19     end;
20     test := (i > n);
21 end;
22
23 begin
24     read(n, m);
25     for i := 1 to n do
26         read(a[i]);
27     low := 0;
28     high := inf;
29     while high - low > 1 do begin
30         mid := (high + low) div 2;
31         if test(mid) then
32             high := mid
33         else
34             low := mid;
35     end;
36     writeln(high);
37 end.
```

## 6 ★★☆☆☆☆ 享受巧克力

贝西收到了  $N$  块巧克力，她会在接下来的  $D$  天里吃掉这些巧克力，她想制定一个计划，让她每天的快乐度都保持在较高的水平上。

在第一天刚开始的时候，贝西的快乐度为 0。巧克力必须从第一块吃起，不能打乱食用的次序，因为公牛们是按照这个顺序送给她的。吃掉第  $i$  块巧克力，会让她的快乐度立即增加  $A_i$ 。贝西在一天内可以连续食用多块巧克力，也可以一块也不吃。不管如何，当她晚上睡觉之后，快乐度会在半夜减半，如果快乐度是个奇数，则向下取整。请问贝西应该怎样吃巧克力，才能让她在  $D$  天内最小快乐度最大呢？

### 输入格式

- 第一行：两个整数  $N$  和  $D$ ， $1 \leq N, D \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $H_i$ ， $1 \leq H_i \leq 10^6$

### 输出格式

- 单个整数：表示贝西在  $D$  天内的最小快乐度的最大值

### 样例输入

5 5  
10  
40  
13  
22  
7

### 样例输出

24

### 解释

第一天吃前两块巧克力，第二天一块都不吃，接下来三天每天各吃一块。

## 来源

Chocolate Eating, 2010 Feb

原题还要求输出方案，为了方便起见做了修改。

## 代号

ceating

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 50000;
4  int n, d;
5  int h[maxn];
6
7  bool test(long long x) {
8      int i = 0;
9      long long s = 0;
10     for (int k = 1; k <= d; k++) {
11         s = s / 2;
12         while (s < x && i < n) {
13             s += h[i];
14             i++;
15         }
16         if (s < x) return false;
17     }
18     return true;
19 }
20
21 int main() {
22     cin >> n >> d;
23     for (int i = 0; i < n; i++) {
24         cin >> h[i];
25     }
26     long long low = 0;
27     long long high = maxn * 1000000LL;
28     while (high - low > 1) {
29         long long mid = (low + high) / 2;
30         if (test(mid)) {
31             low = mid;
32         } else {
33             high = mid;
34         }
35     }
36     cout << low << endl;
37     return 0;
38 }
```

```
1  const
2      maxn = 50000;
3      inf = maxn * int64(1000000);
4  var
5      n, d, i : longint;
6      low, mid, high : int64;
7      h : array[1..maxn] of longint;
8
9  function test(x:int64):boolean;
10 var
11     i, k : longint;
12     s : int64;
13 begin
14     i := 1;
15     s := 0;
16     for k := 1 to d do begin
17         s := s div 2;
18         while (s < x) and (i <= n) do begin
19             inc(s, h[i]);
20             inc(i);
21         end;
22         if s < x then exit(false);
23     end;
24     exit(true);
25 end;
26
27 begin
28     read(n, d);
29     for i := 1 to n do
30         read(h[i]);
31     low := 0;
32     high := inf;
33     while high - low > 1 do begin
34         mid := (low + high) div 2;
35         if test(mid) then
36             low := mid
37         else
38             high := mid;
39     end;
40     writeln(low);
41 end.
```

## 7 ★★☆☆☆☆ 愤怒的奶牛

约翰有  $N$  间牛棚，这些牛棚坐落在一条直线上，第  $i$  间牛棚位于坐标  $X_i$  的位置。他要把  $C$  头奶牛安排在这些牛棚里。每间牛棚最多可以放一头奶牛，也可以空着。这些奶牛的脾气都很暴躁，所以把它们分得越远越好。请你帮助约翰安排这些奶牛的住处，使得她们两两之间的最短距离最大。

### 输入格式

- 第一行：两个整数  $N$  和  $C$ ， $2 \leq C \leq N \leq 10^5$
- 第二行和第  $N+1$  行：一个整数  $X_i$ ， $0 \leq X_i \leq 10^9$

### 输出格式

- 单个整数：最大的最小距离

### 样例输入

```
5 3
1
2
8
4
9
```

### 样例输出

```
3
```

### 解释

约翰可以把他的三头牛放在 1，4 和 8 的位置上。

来源

Aggressive cows, 2005 Feb

代号

aggr

解答

## 8 ★★☆☆☆☆ 跳石头

奶牛们喜欢参加一种特别的运动——跳石头。它们分别在一条小河的两岸设置了起点和终点，各放了一块石头，起点和终点间的跨度有  $L$  米。然后在河中间放置了  $N$  块石头，这些石头和起点终点处于同一条直线上，第  $i$  块石头距离起点有  $D_i$  米。

游戏的时候，奶牛从起点出发，依次跳过每块石头，最后达到终点。如果两块相邻石头之间的距离太大，有些奶牛就会跳不过去，掉到河里。约翰打算使坏，他想偷偷抽掉其中  $M$  块石头，使得剩余的石头之间最近的距离最大。他不能拿走终点和起点处的石头。请问他该拿走哪些石头，才能让剩下的石头的最短距离最长？

### 输入格式

- 第一行：三个整数  $L$ ,  $N$  和  $M$ ,  $1 \leq L \leq 10^9, 0 \leq M \leq N \leq 50000$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $D_i$ ,  $0 < D_i < L$

### 输出格式

- 单个整数：表示最短距离的最大值

### 样例输入

```
25 5 2
2
14
11
21
17
```

### 样例输出

```
4
```

### 解释

移除任何石头之前，最近的距离是起点到 2，移走 2 和 14 之后，最近的距离是从 17 到 21



## 来源

River Hopscotch, 2006 Dec

## 代号

jump

## 教学要点

- 注意  $n = 0$  时，排序算法如果不当，会造成错误。见 pascal 程序中的注释

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 50000;
4  int n, m, l;
5  int x[maxn+1];
6
7  void swap(int i, int j) {
8      int t = x[i]; x[i] = x[j]; x[j] = t;
9  }
10
11 void sort(int l, int r) {
12     swap(l, (l+r)/2);
13     int j = l + 1;
14     int k = l;
15     for (int i = l+1; i < r; i++) {
16         if (x[i] < x[k]) {
17             swap(i,j);
18             swap(j,k);
19             j++; k++;
20         } else if (x[i]==x[k]) {
21             swap(i,j); j++;
22         }
23     }
24     if (l < k) sort(l, k);
25     if (j < r) sort(j, r);
26 }
27
28 bool test(int t) {
29     int gap = x[0];
30     int k = 0;
31     for (int i = 0; i < n; i++) {
32         if (gap < t) {
33             k++;
34         } else {
35             gap = 0;
36         }
37         gap += x[i+1] - x[i];
38     }
39     // tofix: 不加下面这句竟然也能通过所有数据
40     if (gap < t) {
41         k++;
42     }
43     return (k <= m);
44 }
45
46 int main() {
47     cin >> l >> n >> m;
48     for (int i = 0; i < n; i++) {
```

```
49         cin >> x[i];
50     }
51     sort(0, n);
52     x[n] = l;
53     int low = 0;
54     int high = l + 1;
55     while (high - low > 1) {
56         int mid = (low + high) / 2;
57         if (test(mid)) {
58             low = mid;
59         } else {
60             high = mid;
61         }
62     }
63     cout << low << endl;
64     return 0;
65 }

```

```
1  const
2      maxn = 50000;
3  var
4      n, m, l, i, low, high, mid : longint;
5      x : array[1..maxn+1] of longint;
6
7  procedure swap(i,j:longint);
8  var
9      t : longint;
10 begin
11     t := x[i]; x[i] := x[j]; x[j] := t;
12 end;
13
14 procedure sort(l,r:longint);
15 var
16     i, j, k : longint;
17 begin
18     { error when l = 1, r = 0}
19     swap(l, (l+r) div 2);
20     j := l + 1;
21     k := l;
22     for i := l + 1 to r do
23         if x[i] < x[k] then begin
24             swap(i, j); swap(j, k);
25             inc(j); inc(k);
26         end
27     else if x[i] = x[k] then begin
28         swap(i, j);
29         inc(j);
30     end;
```

```

31   if l < k - 1 then sort(l, k-1);
32   if j < r then sort(j, r);
33 end;
34
35 function test(t:longint):boolean;
36 var
37   gap, i, k : longint;
38 begin
39   gap := x[1];
40   k := 0;
41   for i := 1 to n do begin
42     if gap < t then
43       inc(k)
44     else
45       gap := 0;
46     inc(gap, x[i+1] - x[i]);
47   end;
48   if gap < t then inc(k);
49   test := k <= m;
50 end;

```

```

51
52 begin
53   read(l, n, m);
54   for i := 1 to n do
55     read(x[i]);
56   if n > 1 then
57     sort(1, n);
58   x[n+1] := 1;
59   low := 0;
60   high := 1 + 1;
61   while high - low > 1 do begin
62     mid := (low + high) div 2;
63     if test(mid) then
64       low := mid
65     else
66       high := mid;
67   end;
68   writeln(low);
69 end.

```

## 9 ★★★☆☆ 破坏阴谋

约翰的牧场里有  $N$  台机器，第  $i$  台机器的工作能力为  $A_i$ 。保罗阴谋破坏一些机器，使得约翰的工作效率变低。保罗可以任意选取一段编号连续的机器，使它们停止工作。但这样的破坏只能搞一次，而且保罗无法破坏第一台或最后一台机器。请问他该破坏哪些机器才能让剩下机器的工作效率的平均数最小？为了显示存在感，保罗至少必须破坏一台机器。

### 输入格式

- 第一行：单个整数  $N$ ， $3 \leq N \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $1 \leq A_i \leq 10000$

### 输出格式

- 单个浮点数：表示保尔破坏机器之后约翰的最低平均工作效率，以四舍五入的方式保留三位小数

### 样例输入

5  
5  
1  
7  
8  
2

### 样例输出

2.667

### 解释

移去 7 和 8，留下 5，1 和 2，平均数是  $8/3$

来源

Sabotage, 2014 Mar

代号

sabotage

解答

$O(N^3)$  的算法

$O(N^2)$  的算法

$O(N \log \alpha)$  的算法

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 100000;
4  int a[maxn];
5  int n, s;
6
7  bool test(double t) {
8      double max = a[1] - t;
9      double pre = 0;
10     for (int i = 1; i < n - 1; i++) {
11         pre += a[i] - t;
12         if (max < pre) {
13             max = pre;
14         }
15         if (pre < 0) {
16             pre = 0;
17         }
18     }
19     return s - n * t - max < 0;
20 }
21
22 int main() {
23     cin >> n;
24     for (int i = 0; i < n; i++) {
25         cin >> a[i];
26         s += a[i];
27     }
28     double low = 0;
29     double high = s;
30     while (high - low > 0.00001) {
31         double mid = (high + low) * 0.5;
32         if (test(mid)) {
33             high = mid;
34         } else {
35             low = mid;
36         }
37     }
38     cout.setf(ios::fixed);
39     cout.precision(3);
40     cout << low << endl;
41     return 0;
42 }
```

```
1  const
2      maxn = 100000;
3  var
4      a : array[1..maxn] of longint;
5      n, i, s : longint;
6      high, low, mid : double;
7
8  function test(t:double):boolean;
9  var
10     i : longint;
11     max, pre : double;
12 begin
13     max := a[2] - t;
14     pre := 0;
15     for i := 2 to n - 1 do begin
16         pre := pre + a[i] - t;
17         if max < pre then
18             max := pre;
19         if pre < 0 then
20             pre := 0;
21     end;
22     test := s - n * t - max < 0;
23 end;
24
25 begin
26     read(n);
27     for i := 1 to n do begin
28         read(a[i]);
29         inc(s, a[i]);
30     end;
31     low := 1;
32     high := s;
33     while high - low > 0.00001 do begin
34         mid := (high + low) * 0.5;
35         if test(mid) then
36             high := mid
37         else
38             low := mid;
39     end;
40     writeln(low:0:3);
41 end.
```

## Chapter 3

### 堆

## 10 ★★☆☆☆☆ 锯栅栏

约翰正在修复农场的一段围栏。进过丈量，他需要  $N$  块木板，第  $i$  块木板的长度应该是  $L_i$ 。他订购了一根很长的木料，长度就是这  $N$  块木板的总长度，由于约翰没有锯子，所以他找到了老唐，想让他帮忙锯开木料。

老唐是个生意精，他乐意代劳，但劳动不是无偿的。每开动一次锯子，可以把木料锯成两段，锯出  $N$  块木板就需要开动  $N - 1$  次机器。老唐说，每次开机都要收加工费，加工费用正比于需要锯开的木料长度。比如要把一段长 20 米的木料锯开，就要收 20 元。

约翰不得不接受了老唐的要求，但他发现不同的切割次序会产生不同的费用。请帮约翰设计一个锯开木料的方案，使得约翰付出的加工费之和最小。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 20000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $L_i$ ， $1 \leq L_i \leq 50000$

### 输出格式

- 单个整数：表示约翰付出的最小费用

### 样例输入

3  
8  
5  
8

### 样例输出

34

### 解释

将 21 锯成 13 和 8，然后再将 13 锯成 8 和 5，总费用为  $21 + 13 = 34$



## 来源

Fence Repair, 2006 Nov

## 代号

plank

## 相关题目

- 果子合并, 2004 年 NOIP 提高组

## 教学要点

1. 这是一道引出堆概念的好题。只谈堆的基本操作, 不需要额外的排序工作, 主过程简单清晰, 适合给普及组水平讲授
2. 由于存在递归过程, 以教师指导学生静态差错为主, 不建议放手让初学者自行调试, 效果不佳
3. 注意数据范围, 答案有超过 32 位的可能
4. 整个过程在两小时左右

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 20000;
4  int q[maxn+1];
5  int s;
6
7  void swap(int i, int j) {
8      int t = q[i]; q[i] = q[j]; q[j] = t;
9  }
10
11 void up(int i) {
12     int j = i / 2;
13     if (j >= 1 && q[j] > q[i]) {
14         swap(i, j);
15         up(j);
16     }
17 }
18
19 void down(int i) {
20     int j = i + i;
21     if (j + 1 <= s && q[j] > q[j+1]) {
22         j++;
23     }
24     if (j <= s && q[j] < q[i]) {
25         swap(i, j);
26         down(j);
27     }
28 }
29
30 int pop(void) {
31     int ret = q[1];
32     q[1] = q[s];
33     s--;
34     down(1);
35     return ret;
36 }
37
38 void push(int x) {
39     s++;
40     q[s] = x;
41     up(s);
42 }
43
44 int main() {
45     int n;
46     cin >> n;
47     for (int i = 0; i < n; i++) {
48         int x;
49         cin >> x;
50         push(x);
51     }
52     long long ans = 0;
53     for (int i = 0; i < n - 1; i++) {
54         int x = pop() + pop();
55         ans += x;
56         push(x);
57     }
58     cout << ans << endl;
59     return 0;
60 }
```

```

1  const
2    maxn = 20000;
3  var
4    q : array[1..maxn] of longint;
5    n, s, i, x : longint;
6    ans : int64;
7
8  procedure swap(i, j : longint);
9  var
10    t : longint;
11  begin
12    t := q[i]; q[i] := q[j]; q[j] := t;
13  end;
14
15  procedure up(i : longint);
16  var
17    j : longint;
18  begin
19    j := i div 2;
20    if (j >= 1) and (q[j] > q[i]) then begin
21      swap(i, j);
22      up(j);
23    end;
24  end;
25
26  procedure push(x : longint);
27  begin
28    inc(s);
29    q[s] := x;
30    up(s);
31  end;
32
33  procedure down(i : longint);

```

```

34  var
35    j : longint;
36  begin
37    j := i + i;
38    if (j + 1 <= s) and (q[j+1] < q[j]) then
39      j := j + 1;
40    if (j <= s) and (q[i] > q[j]) then begin
41      swap(i, j);
42      down(j);
43    end;
44  end;
45
46  function pop : longint;
47  begin
48    pop := q[1];
49    q[1] := q[s];
50    dec(s);
51    down(1);
52  end;
53
54  begin
55    read(n);
56    for i := 1 to n do begin
57      read(x);
58      push(x);
59    end;
60    for i := 1 to n - 1 do begin
61      x := pop() + pop();
62      push(x);
63      inc(ans, x);
64    end;
65    writeln(ans);
66  end.

```

## 11 ★★★☆☆ 奶牛探险

奶牛们驾驶着一辆卡车，在丛林中探险。卡车每前进 1 公里会消耗 1 升汽油，奶牛必须赶在汽油用尽之前返回城市。奶牛目前距离城市有  $L$  公里，但邮箱里只有  $P$  升汽油了。

好在回去的路上存在  $N$  个汽油补给点，第  $i$  个补给点距离终点有  $X_i$  公里，可以补给  $U_i$  升汽油。这辆卡车的油箱非常大，多少油都能装得下。丛林是个危险的地方，除非必要，奶牛们是不太希望停车加油的。请算出奶牛至少要加几次油才能到达终点，或者告诉它们无论怎么样都到不了目的地。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10000$
- 第二行到第  $N+1$  行：第  $i+1$  行有两个整数  $X_i$  和  $U_i$ ， $1 \leq X_i \leq L, 1 \leq U_i \leq 100$
- 第  $N+2$  行：两个整数  $L$  和  $P$ ， $1 \leq P \leq L \leq 10^6$

### 输出格式

- 单个整数：为达到终点最少停车加油的次数，如果无法到达终点，输出  $-1$

### 样例输入

```
4
4 4
5 2
11 5
15 10
25 10
```

### 样例输出

```
2
```

### 解释

前进 10 公里，加 10 升油，再前进 4 公里，加 5 升油，就能直接开到终点了

来源

Expedition, 2005 Open

代号

exp

## 12 ★★★☆☆ 赞助学费

人类可以上大学，而奶牛们却没学可上。为解决这个问题，贝西和她的伙伴们创立了一所奶牛大学，取名为哞哞大学。

今年，共有  $C$  头奶牛申请入学，第  $i$  头奶牛的考试分数为  $S_i$ ，没有两头奶牛的分数是一样的。奶牛们必须依赖奖学金才能进入大学学习，第  $i$  头奶牛申请的奖学金为  $Q_i$  元。贝西负责选拔新生的工作。首先，今年只能招  $N$  头奶牛， $N$  是个奇数。本着一个都不能少的原则，不能让入学的奶牛数量少于  $N$ 。其次，由于学校的助学基金只能支付  $F$  元，所以她选拔的新生所申请的奖学金数量之和不能超过  $F$ 。

在满足这两个条件的前提下，贝西希望新生的考试分数的中位数越高越好。所谓中位数，就是按大小排序后处在中间位置的分数，比如  $\{3, 8, 9, 7, 5\}$  的中位数是 7。请帮助贝西确定接受哪些奶牛的申请才可以使学校新生的考试分数的中位数达到最大。

### 输入格式

- 第一行：三个整数  $N$ ， $C$  和  $F$ ， $1 \leq N \leq 19999 \leq C \leq 10^5$
- 第二行到第  $C+1$  行：第  $i+1$  行有两个整数  $S_i$  和  $Q_i$ ， $0 \leq S_i \leq 2 \times 10^9, 0 \leq Q_i \leq 10^5$

### 输出格式

- 单个整数：表示考试分数的最大中位数，如果预算不够资助任意  $N$  头奶牛的组合，输出  $-1$

### 样例输入

```
3 5 70
30 25
50 21
20 20
5 18
35 30
```

### 样例输出

```
35
```

### 解释

贝西接受分数为 5, 35, 50 的奶牛，中位数为 35，需支付的奖学金总额为  $18 + 30 + 21 = 69$ ，在预算范围之内

## 来源

Moo University - Financial Aid, 2004 Mar

## 代号

finance

## 13 ★★★★★☆ 奶牛优惠券

约翰需要买更多的奶牛！市场上有  $N$  头奶牛待售，第  $i$  头奶牛的原价是  $P_i$  元。约翰只有  $M$  元，不过他还有  $K$  张优惠券。如果他在买第  $i$  头牛的时候使用一张优惠券，那么就可以享受一个折扣价  $C_i$ 。每头牛只能使用一次优惠券，请问约翰最多能买多少头奶牛回家？

### 输入格式

- 第一行：三个整数  $N$ ， $K$  和  $M$ ， $1 \leq K \leq N \leq 50000$ ， $1 \leq M \leq 10^{14}$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $P_i$  和  $C_i$ ， $1 \leq C_i \leq P_i \leq 10^9$

### 输出格式

- 单个整数：表示约翰最多能买几头牛

### 样例输入

```
4 1 7
3 2
2 2
8 1
4 3
```

### 样例输出

```
3
```

### 解释

将优惠券用在第三头奶牛上，可以用 6 元买下前三头



来源

Cow Coupons, 2012 Feb

代号

coupons

## 14 ★★★★★☆ 都市地平线

约翰带着奶牛去都市观光。在落日的余晖里，他们看到了一幢接一幢的摩天高楼的轮廓在地平线上形成美丽的图案。以地平线为  $X$  轴，每幢高楼的轮廓是一个位于地平线上的矩形，彼此间可能有重叠的部分。奶牛一共看到了  $N$  幢高楼，第  $i$  幢楼的高度是  $H_i$ ，两条边界轮廓在地平线上的坐标是  $A_i$  到  $B_i$ 。请帮助奶牛们计算一下，所有摩天高楼的轮廓覆盖的总面积是多少。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 40000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有三个整数  $A_i$ ， $B_i$  和  $H_i$ ， $1 \leq A_i < B_i \leq 10^9$ ， $1 \leq H_i \leq 10^9$

### 输出格式

- 单个整数：表示摩天高楼轮廓所覆盖的总面积

### 样例输入

```
4
2 5 1
9 10 4
6 8 2
4 6 3
```

### 样例输出

```
16
```

### 解释

只有第一幢楼和最后一幢楼有 1 个单位的重叠面积

来源

City Horizon, 2007 Open

代号

horizon

## Chapter 4

# 数值计算

## 15 ★☆☆☆☆ 负二进制

奶牛开始设计新的计算机了，很自然，它们想到了要用二进制表示数字。但它们不知道补码，所以在表示负数的时候遇到了麻烦。然而，它们想出了另一种解决方法。奶牛的新方案是使用  $-2$  作为进制的基数，每个数位的权重依次为  $1, -2, 4, -8, 16, -32$  等等。这种新式的进制可以保证每个整数都有唯一的表示，例如  $3$  的表示为

$$3 = (-2)^2 + (-2)^1 + (-2)^0 = (111)_{-2}$$

负进制的最大好处是，表示负数不需要用负号，例如  $-9$  可以表示成

$$-9 = (-2)^3 + (-2)^1 + (-2)^0 = (1011)_{-2}$$

来看看更多的例子。 $1$  到  $7$  的表示依次是

$$(1)_{-2}, (110)_{-2}, (111)_{-2}, (100)_{-2}, (101)_{-2}, (11010)_{-2}, (11011)_{-2}$$

$-1$  到  $-7$  的表示依次是

$$(11)_{-2}, (10)_{-2}, (1101)_{-2}, (1100)_{-2}, (1111)_{-2}, (1110)_{-2}, (1001)_{-2}$$

请你帮助奶牛把一个十进制数转成负二进制吧。

### 输入格式

- 第一行：单个整数  $N$ ， $-2 \times 10^9 \leq N \leq 2 \times 10^9$

### 输出格式

- 一个整数： $N$  的负二进制表示，不要在头部出现多余的  $0$

### 样例输入

-13

### 样例输出

110111

## 来源

The Moronic Cowmpouter, 2006 Feb

## 代号

negtwo

## 相关题目

- 1998 年第五届波兰信息学竞赛（POI V）决赛（III Stage, Finals）第一题——“ATM'S”
- 2000 年 NOIP 提高组复赛第一题——“进制转换”

## 解答

先定低位。

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3
4  void print(int n) {
5      int r;
6      if (n % 2 == 0) {
7          r = 0;
8      } else {
9          r = 1;
10     }
11     n = -(n - r) / 2;
12     if (n != 0) {
13         print(n);
14     }
15     cout << r;
16 }
17
18 int main() {
19     int n;
20     cin >> n;
21     print(n);
22     cout << endl;
23     return 0;
24 }
```

```
1  procedure print(n:longint);
2  var
3      r : longint;
4  begin
5      if n mod 2 = 0 then
6          r := 0
7      else
8          r := 1;
9      n := -(n-r) div 2;
10     if n <> 0 then
11         print(n);
12     write(r);
13 end;
14
15 var
16     n : longint;
17
18 begin
19     read(n);
20     print(n);
21     writeln;
22 end.
```

## 16 ★★☆☆☆☆ 分数变小数

请帮助奶牛写个程序，把分数  $N/D$  化成十进制小数形式。用圆括号表示循环节，如果这个小数没有小数部分，以 0 代替。例如：

$$\begin{aligned}1/3 &= 0.(3) \\ 22/5 &= 4.4 \\ 1/7 &= 0.(142857) \\ 2/2 &= 1.0 \\ 3/8 &= 0.375 \\ 45/56 &= 0.803(571428)\end{aligned}$$

### 输入格式

- 第一行：两个整数  $N$  和  $D$ ， $1 \leq N, D \leq 10^5$

### 输出格式

- 一串数字： $N/D$  的小数表示

### 样例输入

45 56

### 样例输出

0.803(571428)



## 来源

Fractions to Decimals, 1993 Qualifying Round

## 代号

fracdec

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxb = 1000000;
4  int d[maxb], first[maxb];
5  int main() {
6      int a, b;
7      cin >> a >> b;
8      cout << a / b << ".";
9      int r = a % b;
10     int l = 0;
11     while (r > 0 && first[r] == 0) {
12         l++;
13         first[r] = l;
14         d[l] = (r * 10) / b;
15         r = (r * 10) % b;
16     }
17
18     if (r == 0) {
19         if (l == 0) {
20             cout << 0;
21         } else {
22             for (int i = 1; i <= l; i++) {
23                 cout << d[i];
24             }
25         }
26     } else {
27         for (int i = 1; i < first[r]; i++)
28             cout << d[i];
29         cout << "(";
30         for (int i = first[r]; i <= l; i++)
31             cout << d[i];
32         cout << ")";
33     }
34     cout << endl;
35     return 0;
36 }
```

```
1  const
2      maxb = 100000;
3  var
4      a, b, r, l, i : longint;
5      d, first : array[1..maxb] of longint;
6  begin
7      read(a, b);
8      write(a div b, '.');
9      r := a mod b;
10     while (r > 0) and (first[r] = 0) do begin
11         l := l + 1;
12         first[r] := l;
13         d[l] := r * 10 div b;
14         r := r * 10 mod b;
15     end;
16     if r = 0 then begin
17         if l = 0 then
18             write(0)
19         else
20             for i := 1 to l do
21                 write(d[i]);
22     end else begin
23         for i := 1 to first[r] - 1 do
24             write(d[i]);
25         write('(');
26         for i := first[r] to l do
27             write(d[i]);
28         write(')');
29     end;
30 end.
```

## 17 ★★☆☆☆☆ 数的幂次

贝西正在学习算术。她的回家作业是计算  $N^P$ 。例如：

$123456^{88} =$   
1129987770413559019467963153621658978635389622595924947762339599136126  
3387265547320084192414348663697499847610072677686227073640285420809119  
1376617325522768826696494392126983220396307144829544079751988205731569  
1498433718478969549886325738202371569900214092289842856905719188890170  
0772424218248094640290736200969188059104939824466416330655204270246371  
3699112106518584413775333247720509274637795508338904731884172716714194  
40898407102819460020873199616

请写个程序，帮助贝西完成作业吧。

### 输入格式

- 第一行：两个整数  $N$  和  $P$ ， $1 \leq N \leq 10^9, 1 \leq P \leq 20000$

### 输出格式

- 一串数字：表示  $N^P$

### 样例输入

2 15

### 样例输出

32768

来源

Cruel Math Teacher, 2009 Feb

代号

power

解答

## Chapter 5

# 排序

## 18 ★★☆☆☆☆ 重排干草

约翰订购了很多干草，他在农场里标记了  $N$  个位置。这些位置近似地构成一个圆环。他原打算让送货司机在  $i$  号位卸下  $B_i$  捆干草。然而，送货司机搞乱了约翰的部署，胡乱卸货之后就离开了。约翰数了数，目前在  $i$  号位有  $A_i$  捆干草， $A_i$  的总和是等于  $B_i$  的，说明司机至少没有少送货。

无奈之下，约翰只能自己来移动这些干草。约翰必须沿相邻位置来移动干草，每移动一捆干草到一个相邻位置，要消耗约翰一单位的能量。请帮约翰规划一下，他最少消耗多少能量才能让所有位置的干草数量从  $\{A_i\}$  变成  $\{B_i\}$ ？由于是圆环，所以 1 号位和  $N$  号位也算作是相邻的。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^5$
- 第二行到  $N + 1$  行：第  $i + 1$  行有两个整数： $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq 1000$

### 输出格式

- 单个整数：表示约翰消耗的最少能量之和

### 样例输入

```
4
7 1
3 4
9 2
1 13
```

### 样例输出

```
13
```

### 解释

将 6 捆干草从 1 号位移到 4 号位，1 捆干草从 3 号位移到 2 号位，6 捆干草从 3 号位移到 4 号位

## 来源

Haybale Restacking, 2012 Mar

2002 年的 NOIP 提高组出了一道“均分纸牌”的问题，是这道题目的简化版本。

## 代号

restack

## 解答

中位数

## 参考程序

```
1  const
2    maxn = 100000;
3  var
4    n, m, i, b : longint;
5    ans : int64;
6    a : array[1..maxn] of longint;
7  procedure sort(l, r : longint);
8  var i, j, k, t : longint;
9  begin
10   i := l; j := r;
11   k := a[(l+r) div 2];
12   repeat
13     while a[i] < k do inc(i);
14     while a[j] > k do dec(j);
15     if i <= j then begin
16       t := a[i]; a[i] := a[j]; a[j] := t;
17       inc(i); dec(j);
18     end;
19   until i > j;
20   if (l < j) and (m <= j) then sort(l, j);
21   if (i < r) and (i <= m) then sort(i, r);
22 end;
23
24 begin
25   read(n);
26   for i := 1 to n do begin
27     read(a[i], b);
28     dec(a[i], b);
29     if i > 1 then
30       inc(a[i], a[i-1]);
31   end;
32   m := (n + 1) div 2;
33   sort(1, n);
34   for i := 1 to n do
35     inc(ans, abs(a[i] - a[m]));
36   writeln(ans);
37 end.
```

## Chapter 6

## 排序（二）



## 19 ★☆☆☆☆ 自私的奶牛

约翰有  $N$  头奶牛，这些奶牛非常自私，不能容忍其他奶牛和自己共享放牧区间。其中第  $i$  头奶牛要求独占  $S_i$  到  $E_i$  的区间。设第  $i$  头奶牛申请的区间为  $(S_i, E_i)$ ，第  $j$  头奶牛申请的区间为  $(S_j, E_j)$ ，那么只有  $E_j \leq S_i$  或  $E_i \leq S_j$  时它们才能和平共处。请问约翰最多可以放多少奶牛同时出来吃草？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $S_i$  和  $E_i$ ， $1 \leq S_i < E_i \leq 10^8$

### 输出格式

- 一个整数：表示最多可以同时放牧几头奶牛

### 样例输入

```
5
2 4
1 12
4 5
7 10
7 8
```

### 样例输出

```
3
```

### 解释

选第一头，第三头和第五头

## 来源

Selfish Grazing, 2009 Dec

## 代号

sgraze

## 解答

## 参考程序

```
1  //cpp todo
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
1  const
2    maxn = 50000;
3  var
4    s, e : array[1..maxn] of longint;
5    n, i, ans, last : longint;
6  procedure sort(l, r : longint);
7  var
8    i, j, k, t : longint;
9  begin
10    i := l; j := r;
11    k := e[(l+r) div 2];
12    while i <= j do begin
13      while e[i] < k do inc(i);
14      while e[j] > k do dec(j);
15      if i <= j then begin
16        t := e[i]; e[i] := e[j]; e[j] := t;
17        t := s[i]; s[i] := s[j]; s[j] := t;
18        inc(i); dec(j);
19      end;
20    end;
21    if l < j then sort(l, j);
22    if i < r then sort(i, r);
23  end;
24
25  begin
26    read(n);
27    for i := 1 to n do
28      read(s[i], e[i]);
29    sort(1, n);
30    for i := 1 to n do
31      if s[i] >= last then begin
32        inc(ans);
33        last := e[i];
34      end;
35    writeln(ans);
36  end.
```

## 20 ★★☆☆☆☆ 录制节目

最近电视里有很多约翰想看的节目。一共有  $N$  个节目，第  $i$  个节目从时刻  $S_i$  开始，到时刻  $T_i$  结束。约翰有两台录像机，每台录像机可以连续工作，录制时间不冲突的节目。如果一个节目的结束时间正好是另一个节目的开始时间，那么这两个节目是可以用一台录像机的。请问约翰用这两台录像机，最多可以录多少不同的节目呢？

### 输入格式

- 第一行：一个整数  $N$ ， $1 \leq N \leq 150$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $S_i$  和  $T_i$ ， $0 \leq S_i < T_i \leq 10^9$

### 输出格式

- 单个整数：表示约翰最多可以录多少节目

### 样例输入

```
6
0 3
6 7
3 10
1 5
2 8
1 9
```

### 样例输出

```
4
```

### 解释

约翰可以用第一台录像机录第一个和第三个节目，用第二台录像机录第二个和第四个节目

来源

Recording the Moolympics, 2014 Jan

代号

recording

解答

## 21 ★★★☆☆ 雄伟的山峦

奶牛们在落基山下避暑，从它们的房子向外望去，可以看到  $N$  座山峰构成的山峦，例如：



奶牛发现每座山峰都是等腰三角形，底边长度恰好是高度的两倍。所以山峰的顶点坐标可由两个底部端点求出。设  $i$  座第山峰的两个底部端点的横坐标分别为  $A_i$  和  $B_i$ 。你能否计算一下这片山峦所覆盖的总面积是多少。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ， $-2^{15} \leq A_i < B_i \leq 2^{15}$

### 输出格式

- 单个整数：表示山峦所占的总面积

### 样例输入

```
5
2 7
6 9
12 15
14 21
20 25
```

### 样例输出

```
114
```

来源

Mountain Majesties, 2002 Open

代号

majesty

## Chapter 7

### 排序（三）

## 22 ★☆☆☆☆ 分数排序

请帮助奶牛按照大小次序，输出所有在 0 到 1 之间的，分母不大于  $N$  既约分数。例如  $N = 5$  时，应该输出

$$\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1}$$

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 500$

### 输出格式

- 按照从小到大的顺序输出分数，每行一个。对每个分数，分子在前，分母在后，中间用字符/分开，

### 样例输入

5

### 样例输出

0/1  
1/5  
1/4  
1/3  
2/5  
1/2  
3/5  
2/3  
3/4  
4/5  
1/1



来源

Ordered Fractions, 1993 Final

代号

ordfrac

解答

## 参考程序

```
1  const int maxn = 500;
2  const int maxm = maxn * maxn / 2;
3  int p[maxn], q[maxn];
4
5  int gcd(int a, int b) {
6      if (b == 0)
7          return a;
8      else
9          return gcd(b, a % b);
10 }
11
12 void swap(int i, int j) {
13     int t = p[i]; p[i] = p[j]; p[j] = t;
14     t = q[i]; q[i] = q[j]; q[j] = t;
15 }
16
17 void sort(int l, int r) {
18     swap(l, (l+r)/2);
19     int j = l;
20     for (int i = l + 1; i < r; i++) {
21         if (p[i]*q[l] < p[l]*q[i]) {
22             j++;
23             swap(i, j);
24         }
25     }
26     swap(l, j);
27     if (j - l > 1) sort(l, j);
28     if (r - j > 2) sort(j + 1, r);
29 }
30
31 int main() {
32     int n;
33     cin >> n;
34     int m = 0;
35     for (int i = 1; i <= n; i++)
36         for (int j = 0; j <= n; j++)
37             if (gcd(i, j) == 1) {
38                 p[m] = j;
39                 q[m] = i;
40                 m++;
41             }
42     sort(0, m);
43     for (int i = 0; i < m; i++) {
44         cout << p[i] << "/" << q[i];
45     }
46     return 0;
47 }
```

```
1  const
2      maxn = 500;
3      maxm = maxn * maxn div 2;
4  var
5      p, q : array[1..maxm] of longint;
6      n, m, i, j : longint;
7  function gcd(a, b : longint) : longint;
8  begin
9      if b = 0 then
10         gcd := a
11     else
12         gcd := gcd(b, a mod b);
13 end;
14
15 procedure swap(i, j:longint);
16 var t : longint;
17 begin
18     t := p[i]; p[i] := p[j]; p[j] := t;
19     t := q[i]; q[i] := q[j]; q[j] := t;
20 end;
21
22 procedure sort(l, r:longint);
23 var i, j : longint;
24 begin
25     swap(l, (l+r) div 2);
26     j := l;
27     for i := l + 1 to r do
28         if p[i]*q[l] < p[l]*q[i] then begin
29             inc(j);
30             swap(i, j);
31         end;
32     swap(l, j);
33     if l < j - 1 then sort(l, j-1);
34     if j + 1 < r then sort(j+1, r);
35 end;
36
37 begin
38     read(n);
39     for i := 1 to n do
40         for j := 0 to i do
41             if gcd(i,j) = 1 then begin
42                 inc(m);
43                 p[m] := j;
44                 q[m] := i;
45             end;
46         sort(1, m);
47     for i := 1 to m do
48         writeln(p[i], '/', q[i]);
49 end.
```

## 23 ★★☆☆☆ 奶牛集体照

约翰有  $N$  头奶牛，准备给它们拍一张集体照。他把奶牛排成一条直线，精心安排了每头奶牛的位置。然而，今天的风儿有点喧嚣，牛群好像传染了熊孩子多动症。一旦约翰离开牛群去按快门，就有几头奶牛乘机乱跑，插到其他奶牛中间。所以照出的相片里，奶牛的位置不一定是约翰想要的样子。约翰很生气，挨个把那些乱动的奶牛教训了一顿，然后把队伍恢复成原来的样子，试图重拍一次。所有被教训过的奶牛都会变得老实，在接下来的所有拍照过程中，再也不会乱动了。但是，在下次拍照的时候，还会有其他奶牛继续这项捣乱事业。这样重复了五次之后，约翰终于崩溃了，他没有精力和这些奶牛玩下去了。你拿到了约翰拍的五张照片，能恢复出约翰原本希望奶牛排列的顺序么？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 20000$
- 第二行： $N$  个整数：表示在第一张照片里的奶牛编号
- 第三行： $N$  个整数：表示在第二张照片里的奶牛编号
- 第四行： $N$  个整数：表示在第三张照片里的奶牛编号
- 第五行： $N$  个整数：表示在第四张照片里的奶牛编号
- 第六行： $N$  个整数：表示在第五张照片里的奶牛编号，所有编号是 1 到  $10^8$  之间的正整数，没有两头奶牛的编号相同

### 输出格式

- 第一行到第  $N$  行：第  $i$  行有一个整数，表示约翰计划中的排列顺序

### 样例输入

```
5
20 30 40 50 10
20 10 30 40 50
30 10 20 40 50
40 10 20 30 50
50 10 20 30 40
```

### 样例输出

```
10
20
30
40
50
```

来源

Cow Photography, 2011 Dec

代号

photo

解答

## Chapter 8

# 归并排序



## 来源

Cow Laundry, 2003 Fall

## 代号

laundry

## 解答

## 25 ★★★★★☆ 奶牛集会

约翰的  $N$  头奶牛每年都会参加“哞哞大会”。哞哞大会是奶牛界的盛事。集会上的活动很多，比如堆干草，跨栅栏，摸牛仔的屁股等等。它们参加活动时会聚在一起，第  $i$  头奶牛的坐标为  $X_i$ ，没有两头奶牛的坐标是相同的。奶牛们的叫声很大，第  $i$  头和第  $j$  头奶牛交流，会发出  $\max\{V_i, V_j\} \times |X_i - X_j|$  的音量，其中  $V_i$  和  $V_j$  分别是第  $i$  头和第  $j$  头奶牛的听力。假设每对奶牛之间同时都在说话，请计算所有奶牛产生的音量之和是多少。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 20000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $V_i$  和  $X_i$ ， $1 \leq V_i \leq 20000, 1 \leq X_i \leq 20000$

### 输出格式

- 单个整数：表示所有奶牛产生的音量之和

### 样例输入

```
4
3 1
2 5
2 6
4 3
```

### 样例输出

```
57
```



## 来源

MooFest, 2004 Open

## 代号

moofest

## 解答

朴素  $O(N^2)$

类似于归并排序的二分  $O(N \log N)$

树状数组  $O(N \log N)$

## Chapter 9

### 贪心——进阶

## 26 ★★☆☆☆☆ 保护花朵

约翰留下了  $N$  只奶牛呆在家里，自顾自地去干活了，这是非常失策的。他还在的时候，奶牛像往常一样悠闲地在牧场里吃草。可是当他回来的时候，他看到了一幕惨剧：他的奶牛跑进了他的花园，正在啃食他精心培育的花朵！约翰要立即采取行动，挨个把它们全部关回牛棚。

约翰牵走第  $i$  头奶牛需要  $T_i$  分钟，因为要算来回时间，所以他实际需要  $2 \cdot T_i$  分钟。第  $i$  头奶牛如果还在花园里逍遥，每分钟会啃食  $D_i$  朵鲜花。但只要约翰抓住了它，开始牵走它的那刻开始，就没法吃花了。请帮助约翰写一个程序来决定押送奶牛的顺序，使得花朵损失的数量最小。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 100000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $T_i$  和  $D_i$ ， $2 \leq T_i \leq 10^6, 1 \leq D_i \leq 100$

### 输出格式

- 单个整数：表示约翰损失的最小花朵数量

### 样例输入

```
6
3 1
2 5
2 3
3 2
4 1
1 6
```

### 样例输出

```
86
```

### 解释

依次制服第六头，第二头，第三头，第四头，第一头和第五头

## 来源

Protecting the Flowers, 2007 Jan

## 代号

flowers

## 解答

## 参考程序

```
1 // cpp todo
```

```
1  const
2      maxn = 100000;
3  var
4      t, d : array[1..maxn] of longint;
5      s, ans : int64;
6      n, i : longint;
7  procedure sort(l, r : longint);
8  var i, j, tk, dk, x : longint;
9  begin
10     i := l; j := r;
11     tk := t[(l+r) div 2];
12     dk := d[(l+r) div 2];
13     while i <= j do begin
14         while (t[i] * dk < d[i] * tk) do inc(i);
15         while (t[j] * dk > d[j] * tk) do dec(j);
16         if i <= j then begin
17             x := t[i]; t[i] := t[j]; t[j] := x;
18             x := d[i]; d[i] := d[j]; d[j] := x;
19             inc(i); dec(j);
20         end;
21     end;
22     if l < j then sort(l, j);
23     if i < r then sort(i, r);
24 end;
25
26 begin
27     read(n);
28     for i := 1 to n do
29         read(t[i], d[i]);
30     sort(1, n);
31     for i := 1 to n do begin
32         inc(ans, s * d[i]);
33         inc(s, t[i]);
34     end;
35     writeln(ans + ans);
36 end.
```

## 27 ★★☆☆☆ 奶牛杂技

奶牛打算逃出牧场，去马戏团冒险。可它们很快发现由于它们的蹄子太笨拙，根本无法站稳在钢丝上。它们还尝试过把自己装在大炮里发射出去，然而可想而知，结局是悲惨的。最终，它们决定练习一种比较简单的杂技：叠罗汉。

表演时会有  $N$  头奶牛出场，第一头牛站在地上，剩下的奶牛挨个爬上到前一头的背上，直到最后一头奶牛爬到最高的地方为止。每头被压在下面的奶牛都会受到来自上方的压力，这是有点危险的。设第  $i$  头奶牛的重量为  $W_i$ ，力量为  $S_i$ 。每头奶牛的压力指数定义为它所承受重量之和与它自己的力量之差。请告诉奶牛们，它们该怎么安排叠罗汉的顺序，才能使得团队中最大的压力指数最小。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $W_i$  和  $S_i$ ， $1 \leq W_i \leq 10^4, 1 \leq S_i \leq 10^9$

### 输出格式

- 单个整数：表示最大压力指数的最小值

### 样例输入

```
3
10 3
2 5
3 3
```

### 样例输出

```
2
```

### 解释

让重量为 10 的奶牛垫底，它的压力指数为  $2 + 3 - 3 = 2$ ，其余两头牛的压力指数都比它小

来源

Cow Acrobats, 2005 Nov

代号

acrobat

解答

## 28 ★★★★★ 二道工序

每天早晨，约翰的  $N$  头奶牛都要排队接受挤奶。为了提高效率，约翰把挤奶过程分成了两道工序。约翰负责第一道工序，第二道工序则让他的好友罗伯帮忙。奶牛们进入牛棚后，首先在约翰处排队，依次接受第一道工序，完成后再去罗伯处排队，接受第二道工序。挤奶过程中没有奶牛插队或者开小差，约翰和罗伯是工作狂，中途不会休息，直到挤完所有奶牛的牛奶。

约翰发现，如果奶牛的排队顺序不加规划，可能会浪费很多时间。比如一些奶牛的第一道工序很长，那么罗伯就会有很长一段时间无事可做。请你帮助约翰计算一下，应该让奶牛按照什么样的规律排队，才能最节约时间。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 25000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ，分别表示第  $i$  头奶牛接受第一道工序和第二道工序所需要的时间， $1 \leq A_i, B_i \leq 20000$

### 输出格式

- 单个整数：表示最少需要多少时间才能挤完所有奶牛的牛奶

### 样例输入

```
3
2 2
7 4
3 5
```

### 样例输出

```
16
```

### 解释

排队顺序应该是第三头，第一头，第二头

来源

The Milk Queue, 2006 Open

代号

mqueue

解答



## Chapter 10

### 栈

## 29 ★☆☆☆☆ 括号序列

括号序列是由左括号“(”和右括号“)”构成的字符串。平衡的括号序列要求“(”和“)”出现的次数一样多，而且每一个前缀中的“(”的出现次数都不少于“)”。最近，奶牛们定义了一种为平衡的括号序列计算分数的规则。

- 首先，如果只有一对括号 $()$ ，则只算 1 分；
- 其次，如果字符串  $A$  有  $s$  分，那么字符串  $(A)$  有  $2s$  分；
- 最后，如果字符串  $A$  有  $s$  分，字符串  $B$  有  $t$  分，那么字符串  $AB$  有  $s+t$  分；

给定一个平衡的括号序列，请帮助奶牛来计算一下它的分数有多少吧，由于数字可能很大，只要输出答案模 12345678910 的余数即可。

### 输入格式

- 第一行：单个整数  $N$ ， $2 \leq N \leq 10^5$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个字符  $c$ ， $c \in \{0,1\}$ ，0 代表左括号，1 代表右括号
- 保证输入所表示的括号序列一定是平衡的

### 输出格式

- 单个整数：表示括号序列的分数模 12345678910 的余数

### 样例输入

6  
0  
0  
1  
1  
0  
1

### 样例输出

3

### 解释

$((()))$  的分数为  $2 \times 1 + 1 = 3$

来源

Best Parenthesis, 2011 Feb

代号

paren

解答

### 30 ★☆☆☆☆ 向右看齐

$N$  头奶牛被紧急动员起来了，它们排成了一条长列。从左向右看，排在第  $i$  个位置的奶牛身高为  $H_i$ 。约翰一声令下，所有奶牛向右看齐。假设每头奶牛只能看到比自己高的牛。请问它们各自看到的最近的一头奶牛分别是谁呢？

#### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^6$
- 第二行到  $N + 1$  行：第  $i + 1$  行有一个整数  $H_i$ ， $1 \leq H_i \leq 10^6$

#### 输出格式

- 第一行到第  $N$  行：第  $i$  行有一个整数  $C_i$ ，表示第  $i$  头奶牛向右看到的最近的一头奶牛编号，如果看不到任何奶牛， $C_i$  为 0

#### 样例输入

6  
3  
2  
6  
1  
1  
2

#### 样例输出

3  
3  
0  
6  
6  
0

来源

Look Up, 2009 Mar

代号

lookup

解答

## Chapter 11

# 队列

### 31 ★★☆☆☆☆ 失误的滑雪

贝西正在参加一项滑雪比赛。她从起点出发的时候，速度恒定为每秒 1 米。然而，随着比赛进程的增加，她会犯很多错误，每次失误都会使她的速度下降。当她第一次失误后，速度会下降到每秒  $1/2$  米，第二次失误后，速度会下降到每秒  $1/3$  米，第  $k$  次失误后，速度会下降到每秒  $1/(k+1)$  米。

约翰记录了贝西的所有失误，一共有  $Q$  个。有两种失误，一种发生在比赛开始后的某个时间点，另一种发生在赛道的某个位置上。有时，贝西可能在某个时间点到达某个位置，而恰好在这个时间点和位置上都有一失误的记录，这两个记录要算作不同的失误，会对贝西的速度造成两次影响。比赛的终点距离起点有 1000 米，请问贝西需要多少时间才能滑过终点？

#### 输入格式

- 第一行：单个整数  $Q$ ， $1 \leq Q \leq 10000$
- 第二行到第  $Q+1$  行：每行开头有个大写字母，代表贝西的一个失误类型：
  - 如果是 T，接下来会有一个整数  $S$ ，表示在比赛开始后的第  $S$  秒钟整发生了一次失误， $1 \leq S \leq 10^7$
  - 如果是 D，接下来会有一个整数  $X$ ，表示在距离起点  $X$  米处发生了一次失误， $1 \leq X \leq 1000$

#### 输出格式

- 单个整数：表示贝西需要多少秒才能滑到终点，如果精确的时间不是整数，则用四舍五入的方法向最接近的整数取整

#### 样例输入

```
2
T 30
D 10
```

#### 样例输出

```
2970
```

#### 解释

前 10 秒，贝西的速度是每秒 1 米，她滑了 10 米。然后她遭遇了第一次失误，在接下来的 20 秒内，她又滑了 10 米。之后她遭遇了第二次失误，还剩下 980 米，所以她共计花去  $10 + 20 + 2940 = 2970$  秒才完成比赛

来源

Bessie Slows Down, 2014 jan

代号

slowdown

解答



## 32 ★★★☆☆ 丑数

给定  $K$  个素数  $P_1, P_2, \dots, P_K$ ，用这些素数作为因数，可以构造无限多个整数，奶牛称这些整数为丑数。例如用 2, 3, 7 生成的丑数从小到大依次为

2, 3, 4, 6, 7, 8, 9, 12, 14, 16, 18, 21, 24, 27, ...

请你帮助奶牛计算这串序列中排在第  $N$  位的数字。

### 输入格式

- 第一行：两个整数  $K$  和  $N$ ， $1 \leq K \leq 100, 1 \leq N \leq 10^6$
- 第二行到第  $K+1$  行：第  $i+1$  行有一个素数  $P_i$ ， $1 < P_i < 2^{31}$

### 输出格式

- 单个整数：代表第  $N$  小的丑数，保证这个数字小于  $2^{31}$

### 样例输入

4 19  
2  
3  
5  
7

### 样例输出

27

## 来源

Humble Numbers, 1997 Championship

## 代号

humble

## 解答

## 相关题目

- panda 的烦恼, 2001 年上海市市队选拔赛 (SHTSC 2001)

## 参考程序

```
1  #include <iostream>
2  using namespace std;
3  const int maxn = 1000000;
4  const int maxk = 100;
5  int q[maxn+1];
6  int p[maxk], d[maxk];
7
8  int main() {
9      int k, n;
10     cin >> k >> n;
11     for (int i = 0; i < k; i++) {
12         cin >> p[i];
13     }
14     q[0] = 1;
15     for (int i = 1; i <= n; i++) {
16         int min = 0;
17         for (int j = 1; j < k; j++) {
18             if (q[d[j]] * p[j] < q[d[min]] * p[min])
19                 min = j;
20         }
21         q[i] = q[d[min]] * p[min];
22         for (int j = 0; j < k; j++) {
23             if (q[d[j]] * p[j] == q[i])
24                 d[j]++;
25         }
26     }
27     cout << q[n] << endl;
28     return 0;
29 }
```

```
1  const
2      maxn = 1000000;
3      maxk = 100;
4
5  var
6      k, n, i, j, min : longint;
7      q : array[0..maxn] of longint;
8      p, d : array[1..maxk] of longint;
9
10 begin
11     read(k, n);
12     for i := 1 to k do
13         read(p[i]);
14     q[0] := 1;
15     for i := 1 to n do begin
16         min := 1;
17         for j := 2 to k do begin
18             if q[d[j]] * p[j] < q[d[min]] * p[min] then
19                 min := j;
20         end;
21         q[i] := q[d[min]] * p[min];
22         for j := 1 to k do begin
23             if q[d[j]] * p[j] = q[i] then
24                 inc(d[j]);
25         end;
26     end;
27     writeln(q[n]);
28 end.
```

### 33 ★★☆☆☆☆ 宴会邀请

约翰正在筹办一个宴会，他想邀请贝西参加。约翰有  $N$  头奶牛，分别以 1 到  $N$  编号，贝西的编号是 1。奶牛形成了  $G$  个圈子。第  $i$  个圈子中有  $S_i$  头不同的奶牛，她们的编号分别是  $A_{i,1}$  到  $A_{i,S_i}$ 。一头奶牛可能同时会出现好几个圈子里。处于同一个圈子的奶牛非常团结，在任何一个圈子里，如果只有一头奶牛没有受到邀请，那么这头被冷落的奶牛就会非常难过。约翰必须邀请贝西参加宴会，请问他最少邀请几头奶牛参加宴会，才不会让任何一头奶牛难过？

#### 输入格式

- 第一行：两个整数  $N$  和  $G$ ， $1 \leq N \leq 10^5$ ， $1 \leq G \leq 10^5$
- 第二行到第  $G+1$  行：第  $i+1$  行首先有一个整数  $S_i$ ，接下来有  $S_i$  个整数，为  $A_{i,1}$  到  $A_{i,S_i}$ ， $1 \leq S_i \leq N$ ，对所有  $1 \leq j \leq S_i$ ， $1 \leq A_{i,j} \leq N$
- 输入数据保证  $S_1 + S_2 + \cdots + S_G \leq 250000$

#### 输出格式

- 单个整数：表示约翰至少需要邀请多少奶牛

#### 样例输入

```
10 4
2 1 3
2 3 4
6 1 2 3 4 6 7
4 4 3 2 1
```

#### 样例输出

```
4
```

#### 解释

由于第一个圈子，必须邀请 3，由于第二个圈子，必须邀请 4，由于第四个圈子，必须邀请 2

来源

Party Invitations, 2013 Jan

代号

invite

解答

## Chapter 12

### 穷举

## 34 ★★☆☆☆☆ 翻转奶牛

$N$  头奶牛排成一行，一些奶牛正面朝向约翰，另一些奶牛背朝约翰，约翰要让所有奶牛都朝向正面。他有一头翻牛机，使用之前，需要固定一个参数  $K$ ， $K$  可以设定为 1 到  $N$  之前的一个整数，一旦设定，就不能再改了。

设定  $K$  后，翻牛机每开动一次，就可以把相邻的  $K$  头奶牛朝向翻面。注意，如果翻牛机上的奶牛少于  $K$  头，它是不能工作的。翻牛机可以工作任意多次，一头奶牛可以被翻动多次。

由于约翰只能选择一个  $K$ ，所以请帮他确定选择哪个  $K$ ，使得他可以使用最少的次数把所有牛翻到正面，记这个最小的次数为  $M$ 。如果有多个  $K$  满足条件，输出其中最小的一个  $K$ 。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 5000$
- 第二行到第  $N+1$  行：第  $i+1$  行只有一个字符  $C_{i+1}$ ：如果  $C_{i+1} = F$ ，表示第  $i$  头奶牛一开始是朝向正面的，如果  $C_{i+1} = B$ ，表示第  $i$  头奶牛一开始是朝向反面的

### 输出格式

- 两个整数： $K$  和  $M$

### 样例输入

7  
B  
B  
F  
B  
F  
B  
B

### 样例输出

3 3

### 解释

选择  $K = 3$ ，将位置 1 到 3，3 到 5，5 到 7 翻转

来源

Face The Right Way, 2007 Mar

代号

cowturn

解答



### 35 ★★☆☆☆☆ 翻转棋

(和 poj 1222 重复)

约翰为奶牛设计了一个智力游戏，名叫翻转棋。翻转棋分为  $N \times M$  个方格，每格两面，一面黑色，一面白色。当奶牛翻动一个格子时，它的颜色就会翻转，由黑变白或由白变黑。如果把所有的格子都翻成白色，就算赢了。

奶牛的蹄子很大，一旦它们翻动某个格子，则和这个格子共享边界的其它格子也会被翻转。约翰给定了奶牛一个初始局面，用  $B_{i,j}$  表示第  $i$  行第  $j$  列的格子颜色， $B_{i,j} = 1$  表示黑色， $B_{i,j} = 0$  表示白色。请帮助奶牛计算一下，把所有格子翻成白色最少需要几步，如果不可能完成，则输出  $-1$ 。

#### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N, M \leq 15$
- 第二行到  $N+1$  行：第  $i+1$  行有  $M$  个整数  $B_{i,1}$  到  $B_{i,M}$ ，对于所有  $1 \leq j \leq M$ ， $0 \leq B_{i,j} \leq 1$

#### 输出格式

- 单个整数，表示最少几步可以把所有格子翻成白色

#### 样例输入

```
4 4
1 0 0 1
0 1 1 0
0 1 1 0
1 0 0 1
```

#### 样例输出

```
4
```

#### 解释

翻动第一列和最后一列的中间两个即可。

来源

Fliptile, 2007 Open

代号

fliptile

解答

## Chapter 13

### 数位问题

## 36 ★★★☆☆ 数数的梦

贝西处于半梦半醒的状态，她意识到自己好像在数自己的小伙伴，一串连续的整数编号从她眼前经过，从  $A$  开始，接下来是  $A+1, A+2, \dots, B-1, B$ 。这些编号以十进制表示，她对每个十进制数字的出现次数很感兴趣。贝西想知道，在整数  $A$  到  $B$  之间，0 到 9 这十个数字，分别出现了多少次？

### 输入格式

- 第一行：两个整数  $A$  和  $B$ ， $1 \leq A \leq B \leq 10^{18}$

### 输出格式

- 十个整数：依次表示 0 到 9 在  $A$  到  $B$  之间出现次数

### 样例输入

129 137

### 样例输出

1 10 2 9 1 1 1 1 0 1

### 解释

1 出现的次数最多，高达十次

来源

Dream Counting, 2006 Dec

代号

dream

解答

### 37 ★★★★★☆ 二进制编号

约翰给每头奶牛用一个数字编号，这些奶牛在集合时，会将自己编号转换成二进制表示，并按照以下规则排队：

- 首先，编号的二进制中 1 出现次数较少的排在队伍的前面；
- 其次，如果 1 的数量一样多，那么编号较小的排在前面；

举个例子，从 4 到 15，有 12 个数字，顺序应该是

100, 1000, 101, 110, 1001, 1010, 1100, 111, 1011, 1101, 1110, 1111

假设编号在  $A$  到  $B$  之间的所有奶牛都在排队，那么这个队伍中排在第  $K$  位的奶牛编号应该是多少呢？

#### 输入格式

- 第一行：单个整数  $A$ ， $0 \leq A < 10^9$
- 第二行：单个整数  $B$ ， $A \leq B < 10^9$
- 第三行：单个整数  $K$ ， $1 \leq K < B - A$

（警告：压缩包里的测试数据不符合目前的描述）

#### 输出格式

- 单个整数：表示所求编号的二进制表示

#### 样例输入

4  
15  
5

#### 样例输出

1001

来源

Cow Queueing, 2003 Dec

代号

cowq

解答

## Chapter 14

# 深度优先搜索



### 38 ★★☆☆☆☆ 象棋皇后

象棋棋盘可分成  $N \times N$  个格子，贝西想在棋盘上放置  $N$  个皇后，它还希望这些皇后不能处于同一行，同一列或者同一条对角线。当  $N = 4$  时，下面两种方法中，左边可行的，而右边的棋子集中在两条对角线上，是不可行的：

	1	2	3	4
1		X		
2				X
3	X			
4			X	

	1	2	3	4
1	X			
2		X		
3				X
4			X	

由于每行有且仅有一个棋子，所以一个放置皇后的布局可用  $N$  个数字来描述，其中第  $i$  个数字表示第  $i$  行的皇后所在的列位置，左上图的布局是：(2, 4, 1, 3)。请你帮助贝西找出所有的可行布局，并输出它们的总数，以及按照字典序排名前三的布局方案。

#### 输入格式

- 单个整数  $N$ ， $4 \leq N \leq 13$

#### 输出格式

- 首先输出字典序排名前三的可行布局，如果不足三个，输出全部布局，每个布局用换行符分割。
- 其次输出一个整数，表示可行布局的总数

#### 样例输入

6

#### 样例输出

2 4 6 1 3 5  
3 6 2 5 1 4  
4 1 5 2 6 3  
4

## 来源

Checker Challenge, 1993 Qualifying Round

## 代号

checker

## 解答

### 39 ★★★☆☆ 拼图游戏

有块拼图被切成了  $R \times C$  块零片，奶牛的任务是把它们正确地恢复成一个矩形。每块零片有四条边，每条边可能是外接边或内接边中的一种。

- 如果一块零片的一条边界是外接边，这条边界不能和其他零片拼接，这些外接边应该构成拼图的最外围轮廓，外接边以数字字符 0 表示。
- 如果一块零片的某条边界是内接边，这条边界必须和其他零片的边界拼接，内接边都有一个小写的英文字母表示类型，只有字母类型相同的两条内接边界才能拼在一起。

因为要看到图画正面的关系，每块零片在拼接时只能通过旋转来调整方向，而不能翻转。约翰保证给定的零片一定能还原拼图，你能帮助奶牛把正确的拼接方法找出来么？

#### 输入格式

- 第一行：两个整数  $R$  和  $C$ ， $1 \leq R, C \leq 10$
- 第二行到第  $RC + 1$  行：第  $i + 1$  行首有一个整数  $K_i$ ，表示该零片的编号， $1 \leq K_i \leq RC$ ，每块零片编号不会相同。接下来会有四个字母，每个字母可能是小写字母或数字 0，依次表示该零片上、右、下、左四条边界的类别

#### 输出格式

- 第一行到第  $RC$  行：在第  $R(i - 1) + j$  行首先有一个数字，代表在第  $i$  行第  $j$  列应放置的零片编号，其次有四个字母，表示该零片上右下左四条边界的字符类别。如有多种解，输出其中第一行零片编号最小的解，如果仍然有多组解，输出其中第二行零片编号最小的解，以此类推

样例输入

2 3  
1 c d 0 0  
2 0 d b 0  
3 c 0 d a  
4 b a b 0  
5 d 0 0 e  
6 0 0 b e

样例输出

1 0 c d 0  
3 0 d a c  
5 0 0 e d  
2 d b 0 0  
4 a b 0 b  
6 e 0 0 b

解释

+---+ +---+ +---+  
| 1 c c 3 d d 5 |  
+-d-+ + a + +-e-+  
  
+-d-+ +-a-+ +-e-+  
| 2 b b 4 b b 6 |  
+---+ +---+ +---+

来源

Jigsaw Puzzles, 2008 Dec

代号

jigsaw

解答

## 40 ★★★☆☆ 埃及分数

古埃及的奶牛会将一个小于 1 的有理数拆成几个单位分数的和，比如

$$\frac{2}{3} = \frac{1}{2} + \frac{1}{6}$$

单位分数就是分子为 1 的分数。拆分后的单位分数必须从大到小书写，其中不能出现相同的单位分数，所以  $1/3 + 1/3$  不算是  $2/3$  的拆分。一个有理数可能有多种拆分方法，最好的拆分方法应该首先让单位分数的项数尽量少，其次应该让每个单位分数的大小应尽量均匀，也就是说最小的单位分数应该尽可能的大。例如对于  $19/45$ ，有以下拆分方法：

$$\begin{aligned}\frac{19}{45} &= \frac{1}{3} + \frac{1}{12} + \frac{1}{180} \\ &= \frac{1}{3} + \frac{1}{15} + \frac{1}{45} \\ &= \frac{1}{3} + \frac{1}{18} + \frac{1}{30} \\ &= \frac{1}{4} + \frac{1}{6} + \frac{1}{180} \\ &= \frac{1}{5} + \frac{1}{6} + \frac{1}{18}\end{aligned}$$

最后一种拆分方法是最好的，因为它的最小分数是  $1/18$ ，比其他表示都要大。请你帮助奶牛找出有理数  $A/B$  的最好的单位分数的拆分吧。可以证明任何一个有理数至少存在一种拆分方法。

### 输入格式

- 第一行：两个整数  $A$  和  $B$ ， $1 \leq A < B \leq 1000$

### 输出格式

- 第一行：单个整数  $N$ ，表示最优拆分方案中单位分数的项数
- 第二行到第  $N+1$  行：第  $i$  行有一个整数  $Q_i$ ，分别表示最优拆分方案中第  $i$  个单位分数的分母
- 保证  $N \leq 5$ ， $Q_1 \times Q_2 \times \cdots \times Q_N < 2^{31}$

### 样例输入

19 45

### 样例输出

3  
5  
6  
18

## 来源

Babylonian Fractions, 1997 Jan

## 代号

egpyt

## 解答

可以证明每个真分数都至少有一个埃及分数的表示。以下提供两种思路：

第一种方法的基本思想是先把  $A/B$  拆成  $A$  个  $1/B$ ，然后设法将重复的  $1/B$  拆成更小的分数，只要反复利用恒等式

$$\frac{1}{K} = \frac{1}{K+1} + \frac{1}{K(K+1)}$$

即可。举个例子：

$$\begin{aligned}\frac{3}{K} &= \frac{1}{K} + \frac{1}{K} + \frac{1}{K} \\ &= \frac{1}{K} + \frac{1}{K+1} + \frac{1}{K(K+1)} + \frac{1}{K+1} + \frac{1}{K(K+1)} \\ &= \frac{1}{K} + \frac{1}{K+1} + \frac{1}{K+2} + \frac{1}{(K+1)(K+2)} + \frac{1}{K(K+1)} \\ &\quad + \frac{1}{K(K+1)+1} + \frac{1}{K(K+1)(K(K+1)+1)}\end{aligned}$$

但这种方法得到的分母太大，不实用。

第二种想法据传最先由斐波那契给出。他的想法是每次总取尽量大的一个单位分数。假设要分解的有理数是  $A/B$ ，找到唯一的一个  $q$ ，使得  $A/B$  介于  $1/q$  和  $1/(q+1)$  之间，即

$$\frac{1}{q} \leq \frac{A}{B} < \frac{1}{q+1}$$

如  $A/B = 1/q$ ，则分解结束。否则，令  $\frac{A'}{B'} = \frac{A}{B} - \frac{1}{q}$ ，用同样的方法分解  $\frac{A'}{B'}$  后，就得到了  $A/B$  的分解式。由于

$$\frac{A'}{B'} = \frac{A}{B} - \frac{1}{q} < \frac{1}{q+1} - \frac{1}{q} = \frac{1}{q(q+1)} < \frac{1}{q}$$

所以  $\frac{A'}{B'}$  一定比  $\frac{1}{q}$  小，这样得到的分解式每项都不同，而且大小是递减的。最后要解决的是，为什么这样的过程一定是有限步的呢？

这是因为实际上， $q$  应该取  $\lceil \frac{B}{A} \rceil$ ，

$$\frac{A'}{B'} = \frac{A}{B} - \frac{1}{q} = \frac{A[B/A] - B}{B[B/A]}$$

最坏情况下， $A[B/A] - B$  和  $B[B/A]$  互素，不能约分，则  $A' = A[B/A] - B$ 。此时我们可以发现  $A' < A$ ，这是因为

$$\frac{A[B/A] - B}{A} = [B/A] - B/A < 1$$

所以依照斐波那契的贪心算法，新得的有理数分子一定小于原来的有理数，所以在分解  $A/B$  的过程中，最多只要执行  $A$  步即可结束。然而，这种做法不见得是最优的，一个反例是按照斐波那契的做法，我们会得到

$$\frac{5}{121} = \frac{1}{25} + \frac{1}{757} + \frac{1}{763309} + \frac{1}{873960180913} + \frac{1}{1527612795642093418846225}$$

而最好的表示其实应该是

$$\frac{5}{121} = \frac{1}{33} + \frac{1}{121} + \frac{1}{363}$$

(内容来自维基百科)



## 41 ★★★★★☆ 栅栏木料

约翰需要一些木料来围住他的牧场，他需要  $M$  根木料，第  $i$  根木料的长度是  $R_i$ 。现在他手上有  $N$  根原木，把原木锯开就得到了木料。第  $i$  根原木的长度为  $L_i$ 。约翰的锯子锋利无比，他技术也不错，所以在切割过程中不会造成浪费，一根原木可以切成任意多段。约翰不能拿几根原木的边角料拼接出一根木料。约翰的原料可能不够切出所有的木料，请你帮他算算，如果只用现在的这些原木，最多能加工出多少根木料？

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 50, 1 \leq M \leq 1023$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $L_i$ ， $1 \leq L_i \leq 10^9$
- 第  $N + 2$  行到第  $N + M + 1$  行：第  $i + N + 1$  行有一个整数  $R_i$ ， $1 \leq R_i \leq 128$

### 输出格式

- 单个整数：表示最多能锯出的木料数量

### 样例输入

```
4 10
30
40
50
25
15
16
17
18
19
20
21
25
24
30
```

### 样例输出

```
7
```

### 解释

锯出的七根木料长度分别为 15, 16, 17, 19, 21, 25, 30。30 和 25 由相等长度的原木得到，21 和 19 由 40 得到，15、16 和 17 由 50 得到

来源

Fence Rails, 1999 Spring

代号

fence

解答

## 42 ★★★★★☆ 解密牛语

约翰的奶牛正在合谋逃出牧场，为此设计了一种加密算法来保护它们的秘密。加密明文时，它们会随机把 C, O, W 三个大写字母插到明文中间，然后交换 C、O 之间的文字和与 O、W 之间的文字。下面是两个例子：

International Olympiad in Informatics  $\implies$  CnOIWternational Olympiad in Informatics

International Olympiad in Informatics  $\implies$  International Cin InformaticsOOlympiad W

为了使解密更复杂，奶牛们会把加密过的密文当做明文再次加密，重复若干次。有一天夜里，约翰截获了一条密文。请你判断一下，它的原文可能是下面这句么？

Begin the Escape execution at the Break of Dawn

注意每个单词之间恰好有一个空格。

### 输入格式

- 第一行：一串字符串，代表密文，长度不超过 75。

### 输出格式

- 单个字符：如果可能，输出“Y”，如果不可能。输出“N”

### 样例输入

Begin the EscCution at the Bre0ape execWak of Dawn

### 样例输出

Y

## 来源

Cryptcowgraphy, 1998 Winter

## 代号

cryptcow

## Chapter 15

# TRIE

## 43 ★★★☆☆ 时间旅行

约翰得到了一台时光机，他可以用这台机器回到过去（但不能到未来），改变他家的牛群。约翰打算依次进行  $N$  步操作，每步操作分为三种：

- 买入操作以  $a$  表示，后接一个参数  $i$ ，表示约翰新买了一头编号为  $i$  的奶牛，将其放入牛棚
- 卖出操作以  $s$  表示，表示约翰会将牛棚中最近买入的一头奶牛卖掉
- 时光倒流的操作以  $t$  表示，后接一个参数  $k$ ，表示约翰将发动时光机，牛棚将返回到第  $k$  步操作之前的状态

请帮助约翰记录牛棚里的奶牛信息，每当执行完一步操作后，输出目前在牛棚里的最近买入的奶牛，如果当时牛棚里没有奶牛了，输出  $-1$ 。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 80000$
- 第二行到  $N + 1$  行：每行代表一步操作，开头有一个小写字母
  - 如果是  $a$ ，表示买入操作，后接一个参数  $i$ ， $1 \leq i \leq 1000000$
  - 如果是  $s$ ，表示卖出操作
  - 如果是  $t$ ，表示时光倒流的操作操作，后接一个参数  $k$ ，若这是第  $i + 1$  行，则  $1 \leq k \leq i$

### 输出格式

- 第一行到第  $N$  行：第  $i$  行有一个整数，表示在第  $i$  步操作后，在牛棚里最近买入的奶牛，如果当时牛棚里没有奶牛，输出  $-1$

样例输入	样例输出	解释
12		
a 5	5	[5]
a 3	3	[5,3]
a 7	7	[5,3,7]
s	3	[5,3]
t 2	5	[5]
a 2	2	[5,2]
t 4	7	[5,3,7]
a 4	4	[5,3,7,4]
s	7	[5,3,7]
t 7	2	[5,2]
s	5	[5]
s	-1	[]

## 来源

Time Travel, 2010 Open

## 代号

ttravel

## 解答

见笔记。



## 44 ★★★★★☆ 一号单词

日常生活中的英文字典在编排过程中，使用的是字典序。字典序的规定空字符串是最小的；对于非空字符串，取决于首字母的大小；首字母相同的两个字符串，则大小取决于排除首字母的部分。

奶牛发现，如果重新定义英文字符的顺序，那么字典的顺序也会改变。给定一本字典中的所有单词，假设奶牛可以任意指定英文字符的顺序，那么有哪些单词可能排在字典的第一名呢？

举例来说，假设有四个单词：omm，moo，mom 和 ommnom，如果奶牛定义 o 在 m 之前，则 omm 可排第一，如果定义 m 在 o 之前，则 mom 可排第一，但余下两个单词是无论如何不可能排在第一的。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 30000$
- 第二行到  $N + 1$  行：每行一个字符串，表示字典中的一个单词，完全由小写字母构成，保证没有两个单词是完全一样的
- 所有单词的长度之和不超过  $3 \times 10^6$

### 输出格式

- 第一行：单个整数  $K$ ，表示有  $K$  个可以排在第一名的单词
- 第二行到  $K + 1$  行：每行输出一个可以排在第一名的单词，表示这些单词的输出顺序必须和输入时一致

### 样例输入

```
4
omm
moo
mom
ommnom
```

### 样例输出

```
2
omm
mom
```

来源

First!, 2012 Dec

代号

first

解答

## Chapter 16

### 并查集

## 45 ★★★☆☆ 真假奶牛

约翰有  $N$  头奶牛，有一部分奶牛是真话奶牛，它们只说真话，而剩下的是假话奶牛，只说假话。有一天，约翰从奶牛的闲谈中陆续得到了  $M$  句话，第  $i$  句话出自第  $X_i$  头奶牛，它会告诉约翰第  $Y_i$  头是一头真话奶牛还是假话奶牛。然而，约翰记性不好，他可能把这些话的内容记错了。请检查一下约翰的记录是否会有矛盾，帮助他找到一个尽量大的  $K$ ，使得约翰记下的前  $K$  句话不矛盾。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 1000, 1 \leq M \leq 10000$
- 第二行到  $M+1$  行：第  $i+1$  行有两个整数： $X_i$  和  $Y_i$ ， $1 \leq X_i, Y_i \leq N$ ，接下来有一个字符：
  - 如果是  $T$ ，表示  $X_i$  说  $Y_i$  是真话奶牛；
  - 如果是  $L$ ，表示  $X_i$  说  $Y_i$  是假话奶牛；

### 输出格式

- 单个整数，即表示题目描述中的  $K$

### 样例输入

```
4 3
1 4 L
2 3 T
4 1 T
```

### 样例输出

```
2
```

### 解释

前两句没有矛盾，但第一句和第三句存在矛盾

## 来源

Liars and Truth Tellers, 2013 Jan

## 代号

truth

## 46 ★★★☆☆ 工作安排

为了维持农场的运转，约翰必须打工赚钱。他接到了  $N$  份工作，每份工作恰好占用他一天的时间。约翰从第一天开始工作，他可以任意安排这些工作的顺序，第  $i$  份工作有  $P_i$  的报酬，但必须在第  $D_i$  天结束之前完成。在截止日期后完成的工作没有报酬。请帮助约翰规划每天的工作，使得他赚到的钱最多。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^5$
- 第二行到  $N + 1$  行：第  $i + 1$  行有两个整数： $D_i$  和  $P_i$ ， $1 \leq D_i, P_i \leq 10^9$

### 输出格式

- 单个整数，表示约翰最多可以赚多少钱

### 样例输入

```
3
2 10
1 5
1 7
```

### 样例输出

```
17
```

### 解释

第一天做第三个任务，第二天做第一个任务

来源

Work Scheduling, 2009 Open

代号

job

## 47 ★★★☆☆ 导航难题

约翰所在的乡村可以看做一个二维平面，其中有  $N$  座牧场，每座牧场都有自己的坐标，编号为 1 到  $N$ 。牧场间存在一些道路，每条道路道路连接两个不同的牧场，方向必定平行于  $X$  轴或  $Y$  轴。乡下地方的道路不会太多，连通两座牧场之间的路径是唯一的。

突然间，约翰的导航仪失灵了，牧场的坐标记录全部消失了。所幸的是，约翰找到了表示道路的数据，可以通过这些信息得知牧场间的相对位置。但贝西有急事，在约翰工作到一半的时候就要知道一些牧场间的曼哈顿距离。这时，如果约翰能从找回的道路信息之间推算出答案，就会告诉贝西。请你帮助约翰来回答贝西的问题吧。 $(x_1, y_1)$  和  $(x_2, y_2)$  间的曼哈顿距离定义为  $|x_1 - x_2| + |y_1 - y_2|$ 。

### 输入格式

- 第一行：两个整数  $N$  和  $Q$ ， $1 \leq N \leq 40000, 1 \leq Q \leq 50000$
- 第二行到  $Q + 1$  行：每行代表一个事件，首先有个大写字母表示事件类型：
  - 如果字母是 E、S、W 和 N 中的一个，表示约翰找回了一条道路的信息，接下来有三个整数  $U$ ， $V$  和  $L$ ， $L$  表示道路的长度， $U, V$  表示道路连接的两个牧场，字母 E、S、W、N 分别表示  $U$  在  $V$  的东、南、西、北方向。 $1 \leq U, V \leq N, 1 \leq L \leq 1000$
  - 如果字母是 D，表示一个询问，接下来会有两个整数  $U$  和  $V$ ，表示询问的是牧场  $U$  和  $V$  间的曼哈顿距离， $1 \leq U, V \leq N$

### 输出格式

- 对每个询问，输出牧场间的曼哈顿距离，如果当时恢复的数据还不足以得知答案，输出  $-1$ 。以换行符分隔答案



### 样例输入

```
7 9
E 1 6 13
D 1 6
E 6 3 9
S 3 5 7
D 1 4
N 4 1 3
W 2 4 20
S 4 7 2
D 2 6
```

### 样例输出

```
13
-1
10
```

### 解释

最后一个询问中，6 号牧场和 2 号牧场间的南北距离为 3，东西距离为 7，故答案为 10

来源

Navigation Nightmare, 2004 Feb

代号

navigate

解答

## 48 ★★★★★☆ 叠积木

约翰和贝西在玩叠积木，每块积木是大小一样的立方体，将几块积木叠在一起可以得到更高的长方体。积木是有磁性的，叠起来的积木分不开。

约翰负责把积木叠高，而贝西负责清点积木的高度。一开始共有  $N$  块积木，编号为 1 到  $N$ ，所有积木都是分开的，没有积木叠在一起。

每次移动积木的时候，约翰会先选择一块积木  $X$ ，再选择另一块积木  $Y$ ，把  $X$  搬到  $Y$  的上方。如果  $X$  已经和其它积木叠在一起了，那么应将这叠积木整体移动到  $Y$  的上方；如果  $Y$  已经和其它积木叠在一起了，假设在  $Y$  上方最高处的积木为  $Z$ ，那么应将  $X$  所在的那叠积木移动到  $Z$  的上方。

在约翰辛苦劳动的同时，贝西在一边计算积木已经叠得多高了。约翰和贝西是交替工作的，可惜贝西不太擅长于数数，请你帮她计算一下，在一些时刻，指定的积木的下方还有多少其他积木。

### 输入格式

- 第一行：两个整数  $N$  和  $Q$ ， $1 \leq N \leq 30000, 1 \leq Q \leq 100000$
- 第二行到  $Q + 1$  行：每行描述一个事件，表示约翰的一个移动操作，或贝西的一个查询，首先有一个大写字母：
  - 如果字母是 M，随后会有两个整数  $X$  和  $Y$ ，约翰把积木  $X$  移动到  $Y$  的上方， $1 \leq X, Y \leq N$ ，保证移动之前  $X$  和  $Y$  不会在同一堆积木里；
  - 如果字母是 C，随后有一个整数  $T$ ，表示贝西想知道在积木  $T$  的下方还有多少其它积木， $1 \leq T \leq N$

### 输出格式

- 对每个来自贝西的查询，输出正确答案，用换行符分隔

### 样例输入

```
6 6
M 1 6
C 1
M 2 4
M 2 6
C 3
C 4
```

### 样例输出

```
1
0
2
```

### 解释

第一次查询时，1 下只有一个 6；第二次查询时，3 下没有任何积木；第三次查询时，4 下有两块积木：1 和 6

来源

Cube Stacking, 2004 Open

代号

cubes

## Chapter 17

### 哈希

## 49 ★★★★★☆ 电灯

牛棚里一共有  $N$  盏灯，电灯间有  $M$  条电线，其中第  $i$  条电线连接了第  $A_i$  和  $B_i$  盏灯， $A_i \neq B_i$ ，也不会有多条电线重复连接同一组灯。每盏灯上有个开关，开关会改变灯的状态——把不亮的灯点亮，或亮的灯变成不亮。不仅如此，由于电线的作用，按下某盏灯的开关后，和这盏灯直接相连的其他电灯也会随之改变状态。

假设刚开始的时候，所有电灯都是关着的，请问按下哪些灯的开关，才能把所有的灯都点亮？输出按下开关的最少次数，保证解法一定存在。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 35, 1 \leq M \leq 595$
- 第二行到第  $M+1$  行：第  $i+1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数：表示最少按下的开关次数

### 样例输入

```
5 6
1 2
1 3
4 2
3 4
2 5
5 3
```

### 样例输出

```
3
```

### 解释

按下第一盏灯，第四盏灯，第五盏灯的开关

来源

Lights, 2009 Nov

代号

lights



## 50 ★★★★★ 宝藏图

麋鹿卡农是贝西的朋友，他生活在广袤的加拿大森林里。卡农有一批宝藏，他要把它埋藏起来。卡农所在的森林里一共有  $N$  个地点，这些地点间有  $N$  条道路，每条道路连接两个地点，都是双向通行的。所有地点彼此间都是连通的，而且从每个地点出发最多，只有四条道路连接。

卡农不喜欢存在四条道路连接的地点，因为他认为十字路口往来的人数太多，容易暴露目标，所以他不会把宝藏藏在这些地方。对剩下的每个地点，卡农会分一部分宝藏埋在那里，并且为这份宝藏画一幅藏宝图，用一个明显的大 X 来标记藏宝的位置。

画完之后，卡农心满意足，他打算将这些藏宝图作为遗产分配给孩子们。但他惊讶地发现，有些藏宝图之间是没有区别的！这是由于藏宝图画得不精确，所以放缩一些道路，改变一些地点间的相对位置，或者把画颠倒，就会发现有些图是等价同构的。图  $A$  和图  $B$  同构的精确定义是需要满足以下两个条件：

- 图  $A$  中的点和图  $B$  中的点一一对应。即存在一个从图  $A$  中的点到图  $B$  中的点的双射  $f$
- 图  $A$  中的边和图  $B$  中的边一一对应：
  - 如果  $u$  和  $v$  在图  $A$  中存在一条边，则  $f(u)$  和  $f(v)$  在图  $B$  中也存在一条边
  - 如果  $u$  和  $v$  在图  $A$  中不存在边，则  $f(u)$  和  $f(v)$  在图  $B$  中也不存在边

你能告诉卡农，他所画的这些地图中，相互不同构的地图还剩多少张么？

### 输入格式

- 第一行：单个整数  $N$ ， $4 \leq N \leq 100000$
- 第二行到第  $N+1$  行：每行两个整数  $U$  和  $V$ ，表示一条道路连接的两个地点编号， $1 \leq U, V \leq N$

### 输出格式

- 单个整数：表示不同构的藏宝地图的数量

### 样例输入

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

### 样例输出

```
3
```

### 解释

每个点都能成为藏宝地点，但在 2 号点和 3 号点埋藏宝藏的话，藏宝图是等价的

来源

Treasure, 2008 Hol

代号

map

## Chapter 18

# 树状数组

## 51 ★★★☆☆ 奶牛抗议

约翰家的  $N$  头奶牛正在排队游行抗议。一些奶牛情绪激动，约翰测算下来，排在第  $i$  位的奶牛的理智度为  $A_i$ ，数字可正可负。

约翰希望奶牛在抗议时保持理性，为此，他打算将这条队伍分割成几个小组，每个抗议小组的理智度之和必须大于或等于零。奶牛的队伍已经固定了前后顺序，所以不能交换它们的位置，所以分在一个小组里的奶牛必须是连续位置的。除此之外，分组多少组，每组分多少奶牛，都没有限制。

约翰想知道有多少种分组的方案，由于答案可能很大，只要输出答案除以 1000000009 的余数即可。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 100000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $-10^5 \leq A_i \leq 10^5$

### 输出格式

- 单个整数：表示分组方案数模 1000000009 的余数

### 样例输入

4  
2  
3  
-3  
1

### 样例输出

4

### 解释

如果分两组，可以把前三头分在一组，或把后三头分在一组；如果分三组，可以把中间两头分在一组，第一和最后一头奶牛自成一组；最后一种分法是把四头奶牛分在同一组里。

## 来源

Generic Cow Protests, 2011 Feb

## 代号

protest

## 解答

方法一：穷举  $O(2^N)$

方法二：动归  $O(N^2)$

方法三：通过调整计算  $f$  的顺序可以将有条件的求和变成无条件的求和。

## 52 ★★★★★☆ 奶牛赛跑

约翰有  $N$  头奶牛，他为这些奶牛准备了一个周长为  $C$  的环形跑牛场。所有奶牛从起点同时起跑，奶牛在比赛中总是以匀速前进的，第  $i$  头牛的速度为  $V_i$ 。只要有一头奶牛跑完  $L$  圈之后，比赛就立即结束了。

有时候，跑得快的奶牛可以比跑得慢的奶牛多绕赛场几圈，从而在一些时刻超过慢的奶牛。这就是最令观众激动的套圈事件了。请问在整个比赛过程中，套圈事件一共会发生多少次呢？

### 输入格式

- 第一行：三个整数  $N$ ， $L$  和  $C$ ， $1 \leq N \leq 10^5, 1 \leq L \leq 25000, 1 \leq C \leq 25000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $V_i$ ， $1 \leq V_i \leq 10^6$

### 输出格式

- 单个整数：表示整个比赛过程中，套圈的次数之和

### 样例输入

```
4 2 100
20
100
70
1
```

### 样例输出

```
4
```

### 解释

两头速度快的奶牛会超过两头速度慢的奶牛各一次

## 来源

Running Laps, 2012 Open

## 代号

running

## 解答

先把所有的奶牛按照速度重新排序，设

$$V_1 \geq V_2 \geq \cdots \geq V_N$$

我们要求的是

$$\sum_{1 \leq i < j \leq N} \left\lfloor \frac{V_i}{V_1} L - \frac{V_j}{V_1} L \right\rfloor = \sum_{1 \leq j \leq N} \sum_{1 \leq i < j} \left\lfloor \frac{V_i}{V_1} L - \frac{V_j}{V_1} L \right\rfloor$$

定义

$$F_j = \sum_{1 \leq i < j} \left\lfloor \frac{V_i}{V_1} L - \frac{V_j}{V_1} L \right\rfloor$$

令

$$\frac{V_i}{V_1} L = A_i + a_i$$

其中  $A_i$  是  $\frac{V_i}{V_1} L$  的整数部分， $a_i$  是  $\frac{V_i}{V_1} L$  的小数部分。

$$\sum_{1 \leq i < j} [A_i - A_j] = \sum_{1 \leq i < j} (A_i - A_j) = \sum_{1 \leq i \leq j} A_i - j \cdot A_j$$

$$\sum_{1 \leq i < j} [a_i - a_j] = \sum_{1 \leq i < j, a_i < a_j} -1$$

## 53 ★★★★★ 道路植草

约翰有  $N$  个牧场，编号为 1 到  $N$ 。它们之间有  $N - 1$  条道路，每条道路连接两个牧场。通过这些道路，所有牧场都是连通的。

刚开始的时候，所有道路都是光秃秃的，没有青草。约翰会在一些道路上批量种草。每次开始种草的时候，约翰会选择一个牧场作为起点，一个牧场作为终点，找到从起点到终点的最短路径，在这条路径上所有的道路上分别种下一棵新的青草。

贝西在监督约翰的工作，她迫不及待地想知道每条道路上已经有多少青草了。约翰的工作总是被贝西打断，他不胜其烦，所以请你来帮忙回答贝西的问题。约翰的工作和贝西的询问是穿插进行的，输入数据将会依次出现  $M$  个事件，以字符 P 开头的事件表示约翰在一条路径上种植了青草，以字符 Q 开头的事件表示贝西在查询一条道路上有多少青草。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $2 \leq N \leq 10^6, 1 \leq M \leq 10^6$
- 第二行到第  $N$  行：第  $i + 1$  行有两个整数  $U_i$  和  $V_i$ ，表示第  $i$  条道路连接的两个牧场编号， $1 \leq U_i, V_i \leq N$
- 第  $N + 1$  行到第  $N + Q$  行：每行首先由一个大写字母：
  - 如果是 P，随后会有两个整数  $A$  和  $B$ ，表示约翰在从  $A$  通向  $B$  的每条道路上都新种了一棵青草， $1 \leq A, B \leq N$
  - 如果是 Q，随后会有两个整数  $A$  和  $B$ ，表示贝西查询一条道路上的青草数量， $A$  和  $B$  是这条道路的两个端点， $1 \leq A, B \leq N$ ，保证输入数据中存在一条  $A$  到  $B$  的道路

### 输出格式

- 对每个查询请求，输出该条道路上青草的数量，以换行符分隔

### 样例输入

```
4 6
1 4
2 4
3 4
P 2 3
P 1 3
Q 3 4
P 1 4
Q 2 4
Q 1 4
```

### 样例输出

```
2
1
2
```



来源

Grass Planting, 2011 Dec

代号

grassplant

## Chapter 19

# 线段树

## 54 ★★★★★☆ 奶牛猜数

为了锻炼思维能力，约翰为奶牛设计了一个游戏。游戏开始前，约翰会在牛棚后面写出  $N$  个整数。约翰和奶牛约定，每个数字在 1 到  $10^9$  之间，没有两个数字是完全一样的。

奶牛的目标是得到关于这串数字尽量多的信息。游戏开始后，奶牛问了  $Q$  个问题，每个问题的格式类似，其中第  $i$  个问题是：

“从第  $L_i$  个数字到第  $R_i$  个数字中，最小的数字是多少？”

对第  $i$  个问题，约翰的回答是  $A_i$ 。因为约翰是个大忽悠，所以奶牛对他的回答半信半疑。请你帮助奶牛判断一下，能否从约翰的回答里发现矛盾，如果有矛盾，他的哪句话最先和之前的话构成了矛盾？

### 输入格式

- 第一行：两个整数  $N$  和  $Q$ ， $1 \leq N \leq 10^5, 1 \leq Q \leq 25000$
- 第二行到第  $N+1$  行：第  $i+1$  行有三个整数  $L_i$ 、 $R_i$  和  $A_i$ ， $1 \leq L_i \leq R_i \leq N, 1 \leq A_i \leq 10^9$

### 输出格式

- 单个整数：如果约翰的回答中找不出矛盾，输出 0，否则输出最先造成矛盾的回答序号

### 样例输入

```
20 4
1 10 7
5 19 7
3 12 8
11 15 12
```

### 样例输出

```
3
```

### 解释

前两个回答可以推导出第五个数到第十个数之间的最小数是 7，第三个回答和这是矛盾的

来源

Haybale Guessing, 2008 Jan

代号

bales

解答

## 55 ★★★★★☆ 跨栏训练

为了让奶牛参与运动，约翰建造了  $K$  个栅栏。每条栅栏可以看做是二维平面上的一条线段，它们都平行于  $X$  轴。第  $i$  条栅栏所覆盖的  $X$  轴坐标的区间为  $[A_i, B_i]$ ， $Y$  轴高度就是  $i$ 。一开始，奶牛在坐标  $(S, K + 1)$  处，它们的家在原点处，所以要想要回家就必须“跨”一些栅栏。

但奶牛们是跨不过栅栏的，它们只能绕过栅栏。在二维平面上，它们只能沿水平和垂直方向移动，如果前进的道路上出现栅栏，它们就不能前进，必须沿水平方向移动到没有栅栏的地方再前进。奶牛们希望走的路越短越好，由于在垂直方向上的路程是确定的，你只需要帮它们求出在水平方向的最短路程就可以了。

### 输入格式

- 第一行：两个整数  $K$  和  $S$ ， $1 \leq K \leq 50000, -10^5 \leq S \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ， $-10^5 \leq A_i \leq B_i \leq 10^5$

### 输出格式

- 单个整数：表示奶牛从起点到终点在水平方向移动的最短总距离

### 样例输入

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

### 样例输出

```
4
```

### 解释

第四个栅栏是最先遇到的，向右移一格绕过它。为了绕过第二个栅栏，再向右移一格，最后为了回到原点向左移两格

来源

Fence Obstacle Course, 2004 Dec

代号

obstacle

解答

## 56 ★★★★★ 预定旅馆

奶牛们准备去苏必利尔湖畔，享受那里的湖光山色以及明媚的阳光。湖边有一家著名旅馆，这家旅馆一共有  $N$  间客房，都在同一楼层，一字顺序排开，只需要拉开窗帘，就能见到波光粼粼的湖面。

旅店前台的工作是忙碌的，他们需要处理订房请求和退房请求。有订房请求的游客会要求预定一段连续的房间。如果能够满足客户的要求，前台总会尽量满足，如果有多处连续的空房可供预定，前台会挑选编号最靠前的优先分配。如果目前没有连续的空房，前台会向他们道歉，请顾客们另找一家宾馆。

同时，前台也会收到退房请求，请求退房时，客户会报出一组连续的区间，前台将这片区域标记成空房。有时，在客户报出的区间中，有些房间本来就是空的，这不要紧，只需要把不是空房的房间全部登记成空房就可以了。

你的工作，就是写一个程序，帮助前台为旅客安排房间。程序一共需要处理  $M$  条请求，在第一个请求之前，所有房间都是空闲的。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 50000, 1 \leq M \leq 50000$
- 第二行到第  $M + 1$  行：每行描述了一个请求，首先有个数字  $T$ ，表示请求类型：
  - 如果  $T = 1$ ，说明这是一个订房请求，接下来会有一个整数  $L$ ，表示顾客需要一段长度为  $L$  的连续空房， $1 \leq L \leq N$
  - 如果  $T = 2$ ，说明这是一个退房请求，接下来会有两个整数  $X$  和  $L$ ， $X$  表示需要办理退房手续的第一个房间号， $L$  表示需要办理退房手续的房间数， $1 \leq X \leq N, 1 \leq L \leq N - X + 1$

### 输出格式

- 对每个订房请求，如果可以满足，输出为它分配的第一个房间号；如果不可满足，输出 0，每个请求的回答用换行符分开

### 样例输入

```
10 6
1 3
1 3
1 3
1 3
2 5 5
1 6
```

### 样例输出

```
1
4
7
0
5
```



来源

Hotel, 2008 Feb

代号

hotel

解答

## 57 ★★★★★ 修路

加拿大只有两种季节——冬天和修路。随着气温增加，修路季节从一个月延长到了六个月。这对于加拿大的麋鹿来说是个头疼的问题。因为它们总是沿着公路移动的。因此，麋鹿卡农想请你设计一套公路监视系统，它能实时跟踪路况，告诉麋鹿们能否移动到想去的城市。

加拿大的城市主要分布在两条东西走向的直线上。整个国家可以分成  $2 \times N$  个方格，每个方格代表一座城市，相邻的城市之间都有公路，但有一些道路可能因为修路的关系暂时不可通行，下面是个例子：

（图缺）

图中每条实线代表目前可以通行的公路，虽说 (2,3) 到 (1,3) 的路暂时关闭了，但绕个远路还是可以通行的。

假设在卡农的监视系统运行之前，所有的道路都是关闭的。你的任务就是帮助卡农的监视系统依次处理  $Q$  个事件，这些事件可能是一份路况报告，也可能是一个查询请求：

- 如果是路况报告，则应以字符 C 或 O 开头，随后是两个城市的坐标： $(r_1, c_1)$  和  $(r_2, c_2)$ ，作为输入数据，保证  $(r_1, c_1)$  和  $(r_2, c_2)$  是两个相邻的城市，C 表示它们之间的公路因施工而暂时关闭，O 表示它们之间的公路因完工而恢复开放；
- 如果是查询请求，则应以字符 A 开头，随后也是两个城市的坐标： $(r_1, c_1)$  和  $(r_2, c_2)$ 。你应该根据目前的路况，回应  $(r_1, c_1)$  和  $(r_2, c_2)$  之间是否可以通行，如果可以，输出 Y，如果不能，输出 N；

### 输入格式

- 第一行：两个整数  $N$  和  $Q$ ， $1 \leq N \leq 15000, 1 \leq Q \leq 50000$
- 第二行到第  $Q+1$  行：第  $i+1$  行有两个整数  $A_i$  和  $B_i$ ， $-10^5 \leq A_i \leq B_i \leq 10^5$

### 输出格式

- 单个整数：对每个查询，如果两城市之间可以通行，输出 Y，否则输出 N，每个回答间用换行符分隔

### 样例输入

缺

### 样例输出

缺

### 解释

来源

Connect, 2007 Open

代号

connect

解答

## Chapter 20

# 平衡树

## 58 ★★★★★ 挑剔的美食家

约翰的奶牛对食物越来越挑剔了。现在，商店有  $M$  份牧草可供出售，奶牛食量很大，每份牧草仅能供一头奶牛食用。第  $i$  份牧草的价格为  $P_i$ ，口感为  $Q_i$ 。约翰一共有  $N$  头奶牛，他要为每头奶牛订购一份牧草，第  $i$  头奶牛要求它的牧草价格不低于  $A_i$ ，口感不低于  $B_i$ 。请问，约翰应该如何为每头奶牛选择牧草，才能让他花的钱最少？

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq M \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq 10^9$
- 第  $N + 2$  行到第  $N + M + 1$  行：第  $i + N + 1$  行有两个整数  $P_i$  和  $Q_i$ ， $1 \leq P_i, Q_i \leq 10^9$

### 输出格式

- 单个整数：表示在满足所有奶牛的要求下的最少总花费，如果不存在这种方案，输出  $-1$ 。

### 样例输入

```
4 7
1 1
2 3
1 4
4 2
3 2
2 1
4 3
5 2
5 4
2 6
4 4
```

### 样例输出

```
12
```

### 解释

第一头牛吃第二份草，花 2 元；第二头牛吃第三份草，花 4 元；第三头牛吃第六份草，花 2 元；第四头牛吃第七份草，花 4 元

来源

Gourmet Grazers, 2007 Dec

代号

gourmet

解答

## 59 ★★★★★☆ 集市班车

奶牛们正在逛集市，它们缺乏锻炼，很快就筋疲力尽了。约翰租了一辆车，准备让奶牛以车代步。集市的地形近似一条直线，从左到右有  $L$  个地点，以 1 到  $L$  编号。奶牛们分成  $Q$  个小组，第  $i$  组有  $M_i$  头奶牛，它们的出发地在  $S_i$ ，目的地在  $T_i$ ，出发地一定在目的地的左边。

约翰租来的班车上  $C$  个位子，最多同时只能装  $C$  头奶牛。问题在于约翰不是土豪，没有多少钱，他的班车只能开一次，即班车只能从 1 开到  $L$ ，中途不能走回头路。由于位子有限，所以一些奶牛就上不了车了。请问，约翰应该搭载哪些奶牛，使得不用走路的奶牛最多？可以只搭载同小组的一部分奶牛。

### 输入格式

- 第一行：三个整数  $Q$ ， $L$  和  $C$ ， $1 \leq Q \leq 50000$ ， $1 \leq L \leq 20000$ ， $1 \leq C \leq 10^9$
- 第二行到第  $Q+1$  行：第  $i+1$  行有三个整数  $S_i$ ， $T_i$  和  $M_i$ ， $1 \leq S_i < T_i \leq L$ ， $1 \leq M_i \leq 10^9$

### 输出格式

- 单个整数：不用走路的奶牛的最大数量

### 样例输入

```
8 15 3
1 5 2
13 14 1
5 8 3
8 14 2
14 15 1
9 12 1
12 15 2
4 6 1
```

### 样例输出

```
10
```

### 解释

把两头奶牛从 1 送到 5，把三头奶牛从 5 送到 8，把两头奶牛从 8 送到 14，把一头奶牛从 9 送到 12，把一头奶牛从 13 送到 14，把一头奶牛从 14 送到 15

## 来源

Fair Shuttle, 2009 Feb

## 代号

shuttle

## 附注

原题中  $C$  小于 100，太少，此外  $L$  可以忽略，做数据时再议

## 解答

想象一下司机比较黑心，可以把坐车坐到一半的奶牛踢下去。

需要一个数据结构，支持取最大，取最小和插入操作，可以用 treap，也可以用两个 heap



## 60 ★★★★★ 奶牛登机（缺）

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 200000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $S_i$  和  $T_i$ ， $1 \leq S_i \leq N, 0 \leq T_i \leq 10^9$
- 保证  $T_1 + T_2 + \cdots + T_N \leq 10^9$

### 输出格式

- 单个整数：表示所有奶牛都坐到座位上的最短时间

### 样例输入

```
3
2 5
3 10
1 5
```

### 样例输出

```
19
```

### 解释

来源

Airplane Boarding, 2013 Feb

代号

boarding

解答

## 61 ★★★★★ 期末总评

贝西参加了  $N$  门考试。第  $i$  门考试的满分为  $P_i$ ，贝西的得分为  $T_i$ 。学校计算总评分的方法是将每门考试的得分之和作为分子，满分之和作为分母，以它们的比值作为总评分。

根据学校的政策，在计算总评分的时候，贝西可以选择取消  $D$  门考试的成绩，这样可以使剩下的考试得到更高的总评分。老师想当然地把贝西得分率最低的  $D$  门考试去掉了（得分率就是一门考试的得分与满分之比）。

然而贝西一眼就看出老师这么做是有问题的。贝西知道，对于不同的  $D$ ，去掉得分率最低的考试不见得是最好的选择。请帮助贝西算算，当  $D$  取 1 到  $N - 1$  之间的整数时，哪些  $D$  可能造成这样惊人的效果？

贝西还注意到一件有意思的事情，她所有考试的得分率都是不同的。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $H_i$  和  $W_i$ ， $0 \leq T_i \leq P_i$ ， $1 \leq P_i \leq 40000$

### 输出格式

- 第一行：单个整数  $K$ ，表示  $D$  有多少种取值方法，会造成老师的选择不是最优的
- 第二行到第  $K + 1$  行：每行一个整数，表示一个让老师犯错的  $D$  的取值，这些值必须以升序输出

### 样例输入

```
5
1 2
5 9
3 8
4 10
1 3
```

### 样例输出

```
2
1
2
```

### 解释

$D = 1$  时，去掉  $1/3$  不如去掉  $3/8$ ； $D = 2$  时，去掉  $1/3$  和  $3/8$  不如去掉  $3/8$  和  $4/10$

## 来源

Cow School, 2007 Jan

## 代号

schul

## 解答

反例构造。假设最优的分数是一个权值很大的满分，另外两次考试都不及格，则应该取权重较小的，而不是分数较高的。

最优值可用二分搜索计算，见日本人书。

## Chapter 21

### 后缀数组

## 62 ★★★★★ 最佳队名

约翰组织他家的  $N$  头奶牛前去报名参加一项集体比赛。奶牛们已经在报名处排成了一字长龙，准备逐个登记姓名。比赛的组织者告诉约翰，在所有参赛队员的名字登记完成之后，这支队伍的名字会自动由全体队员名字的首字母组成。约翰清点了一下奶牛的名字，目前排在第  $i$  名的名字首字母为  $S_i$ ，那么约翰队的名字就应该是  $S_1S_2\ldots S_N$ 。

好在奶牛们还没把名字写到报名表上，所以约翰还可以调整一下报名顺序，使得最终形成的队名尽量在字典序中靠前一些，这样在媒体上的曝光率就可以高一点了。不过，既然奶牛们已经在报名处排成了一条很长的队伍，处于队伍中间的奶牛是动不了的，可以移动到报名台的只有排在队头和队尾的两头奶牛。

约翰可以从队头或队尾中二选一，示意其中一头移动到报名台，在报名表上写下姓名，写完姓名的奶牛会自动离开队伍。如果离开的奶牛原本在队尾，那么原本排在倒数第二的奶牛就成了新的队尾，接下来她也可以来报名处写名字了。如果离开的奶牛原本在队头也是同理。约翰周而复始地指示队头或队尾来报名处写名字，请问他能得到的字典序意义下最靠前的队伍名字是多少呢？

### 输入格式

- 第一行：一个整数  $N$ ， $1 \leq N \leq 30000$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个大写的英文字母  $S_i$

### 输出格式

- 单个整数：表示约翰可以得到的字典序意义下最靠前的队名

### 样例输入

6  
A  
C  
D  
B  
C  
B

### 样例输出

ABCB CD

### 解释

报名顺序应该是队头、队尾、队尾、队尾、队头、队头

来源

Best Cow Line, Gold, 2007 Dec

代号

bclgold

解答

## Chapter 22

# Floodfill



## 63 ★★☆☆☆☆ 池塘计数

最近下大雨，牧场里不少地方都积起了水。牧场可分为  $N \times M$  个方格，一些方格里都是水，变成了池塘，用字符“W”表示，一些方格没水，用字符“.”表示。相邻的方格如果都积水了，会连成一个更大池塘。请问现在牧场里有多少个不连通的池塘？除了在边界上的方格，一个方格应该有八个邻居。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N, M \leq 100$
- 第二行到第  $N+1$  行：第  $i+1$  行有  $M$  个字符，描述了牧场里第  $i$  行的信息，每个字符是“W”或“.”的一种。

### 输出格式

- 单个整数：表示牧场里有多少个不连通的池塘

### 样例输入

```
10 12
W.....WW.
.WWW.....WWW
....WW...WW.
.....WW.
.....W..
..W.....W..
.W.W.....WW.
W.W.W.....W.
.W.W.....W.
..W.....W.
```

### 样例输出

3

### 解释

一个在左上，一个在左下，一个在右边

来源

Lake Counting, 2004 Nov

代号

lkcount

## 64 ★★☆☆☆☆ 卫星图片

约翰得到了他家的卫星图片，图片可分为  $W \times H$  个方格，“\*”表示这个格子是一座建筑，“.”表示这个格子是空地。除了在边界上的格子，每个格子在上下左右四个方向上有邻居，如果两个相邻的格子上都有建筑，那么这两个建筑就是连在一起了。约翰想知道，他家最大的一片连续建筑占据了多少个方格。

### 输入格式

- 第一行：两个整数  $W$  和  $H$ ， $1 \leq W \leq 80, 1 \leq H \leq 1000$
- 第二行到第  $H + 1$  行：第  $i + 1$  行有  $W$  个字符，描述了卫星图片在第  $i$  行地形信息，每个字符是“\*”或“.”的一种。

### 输出格式

- 单个整数：表示最大连续建筑的面积

### 样例输入

```
10 5
..*.....**
.**..*****
.*...*....
..****.***
..****.***
```

### 样例输出

16

### 解释

有三片建筑，大小分别是 4，6 和 16

## 来源

Satellite Photographs, 2005 Nov

## 代号

satpix

## 65 ★★☆☆☆☆ 奶牛万圣节

每年的万圣节，奶牛们都会参加传统的 Trick or Treat 活动。约翰的牧场里有  $N$  间牛棚，到了万圣节那天的晚上，所有奶牛都会出动，去其他牛棚索要糖果。为了维持活动秩序，约翰规定了游行方向，如果奶牛走到了第  $i$  间牛棚，那么接下来它应该去的是第  $X_i$  间牛棚。假设奶牛们都遵循约翰的指挥，请帮助奶牛分别计算一下，从每间牛棚出发，能访问到多少间不同的牛棚？

### 输入格式

- 第一行：单个整数  $N$ ， $4 \leq N \leq 100000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $X_i$ ， $1 \leq X_i \leq N$

### 输出格式

- 第一行到第  $N$  行：每行一个整数，表示从第  $i$  间牛棚出发的奶牛经过的不同牛棚的数量

### 样例输入

4  
1  
3  
2  
3

### 样例输出

1  
2  
2  
3

### 解释

第一间牛棚出发的奶牛只能到自己，第二间和第三间出发的奶牛可以相互走到，第四间奶牛可以到第二、第三间和自己

## 来源

Trick or Treat on the Farm, 2008 Dec

## 代号

treat

## 解答

## Chapter 23

# 最小生成树

## 66 ★★☆☆☆☆ 建造道路

约翰得到了  $N$  片新牧场！他打算把这些牧场连成一片，这需要修建一些道路。牧场间的道路可以修成笔直的直线，第  $i$  片牧场的平面坐标为  $(X_i, Y_i)$ 。幸运的是，这些牧场之间已经存在  $M$  条道路了，约翰可以利用这些道路，不必再修新的。请帮助约翰计算一下，怎么把这些牧场连起来，使得修建的道路总长度最小？

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 1000, 1 \leq M \leq 1000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $X_i$  和  $Y_i$ ， $1 \leq X_i, Y_i \leq 10^6$
- 第  $N + 2$  行到第  $N + M + 1$  行：第  $j + N + 1$  行有两个整数  $A_i$  和  $B_i$ ，表示第  $A_i$  个牧场和第  $B_i$  个牧场之间存在一条可以使用的旧道路， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个浮点数：表示需修建道路的最短长度之和，以四舍五入的方法保留两位小数

### 样例输入

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

### 样例输出

```
4.00
```

### 解释

第一个牧场和第二个牧场之间修建道路，第三个牧场和第四个牧场之间修建道路，两条路的长度都是 2



## 来源

Building Roads, 2007 Dec

## 代号

roads

## 67 ★★☆☆☆ 奶牛施工队

一场地震把约翰家园摧毁了，坚强的约翰决心重建家园。约翰已经修复了  $N$  个牧场，他需要再修复一些道路把它们连接起来。碰巧的是，奶牛们最近也成立了一个工程队，专门从事道路修复。而然，奶牛们很有经济头脑，如果无利可图，它们是不会干的。

约翰和奶牛达成了协议，约翰向奶牛支付  $F$  元，奶牛负责修路，但不必修复所有道路，只要确保所有牧场连通即可。可供奶牛选择修复的道路有  $M$  条，第  $i$  条道路连接第  $U_i$  个牧场和第  $V_i$  个牧场，修复需要  $T_i$  分钟，支出成本为  $C_i$ 。保证连通所有牧场的道路修复方案总是存在的。奶牛们关注的是挣钱速度，即总利润和总施工时间的比值。请帮助奶牛们选择修复哪些道路，才能使它们在单位时间里的利润最大？

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $F$ ， $1 \leq N \leq 400, 1 \leq M \leq 10000, 1 \leq F \leq 2 \times 10^9$
- 第二行到第  $M+1$  行：第  $i+1$  行有四个整数  $U_i$ ， $V_i$ ， $C_i$  和  $T_i$ ， $1 \leq U_i, V_i \leq N, 1 \leq C_i, T_i \leq 2 \times 10^9$

### 输出格式

- 单个浮点数：表示施工队的最大挣钱速度，以四舍五入的方法保留四位小数，如果这个项目一定亏本，输出 0.0000

### 样例输入

```
5 5 100
1 2 20 5
1 3 20 5
1 4 20 5
1 5 20 5
2 3 23 1
```

### 样例输出

```
1.0625
```

### 解释

修最后四条道路，成本总和为 83，时间总和为 16，所以单位时间的利润为  $(100 - 83)/16 = 1.0625$

来源

Earthquake, 2001 Open

代号

quake

## Chapter 24

# 宽度优先遍历

## 68 ★★☆☆☆☆ 拖拉机

约翰忙完了一天的工作，把拖拉机留在牧场上，就回家呼呼大睡去了。谁知奶牛半夜搞恶作剧，从仓库里搬出很多捆干草，把拖拉机团团围住，这下约翰要想开回拖拉机，就要搬掉一些干草了。

约翰的牧场是一块非常广阔没有边界的空地。我们用经纬线将它分割成方格图，每个方格可用一个二维的整数坐标表示。牧场上一共有  $M$  捆干草，第  $i$  捆干草在坐标为  $(X_i, Y_i)$  的方格里，约翰的拖拉机停在坐标为  $(SX, SY)$  的方格里，约翰要把他的拖拉机开回坐标为  $(1, 1)$  的方格。

每堆干草和拖拉机都恰好可以填满一个方格。拖拉机只能向东南西北四个方向前进，不能走对角线。请问约翰最少要移除多少捆干草，才能让他的拖拉机开回  $(1, 1)$ ？注意牧场是无限大的，所以约翰可以将拖拉机开到坐标为负数的地方去。

### 输入格式

- 第一行：三个整数  $N$ ， $SX$  和  $SY$ ， $1 \leq N \leq 5000, 1 \leq SX, SY \leq 1000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $X_i$  和  $Y_i$ ， $1 \leq X_i, Y_i \leq 1000$ ，表示一个存在障碍的格子坐标

### 输出格式

- 单个整数：表示最少需要移除多少捆干草

### 样例输入

```
7 6 3
6 2
5 2
4 3
2 1
7 3
5 4
6 4
```

### 样例输出

```
1
```

### 解释

六捆干草围住了拖拉机，只要移去其中任意一堆障碍就可以了

来源

Tractor, 2012 Mar

代号

tractor

## 69 ★★☆☆☆ 玉米迷宫

约翰带着奶牛们去玩玉米迷宫。迷宫可分成  $N \times M$  个方格，有些方格种了玉米，种了玉米的格子无法通行。贝西一开始在起点，她可以沿上下左右四个方向中移动，每步移动一个方格。在这个迷宫里，有一些神奇的传送点。每个传送点由一对方格组成，并由一个大写的英文字母标示。一旦贝西走入属于传送点的方格，她就会被传送到另外一个方格去。传送点都是双向的，如果贝西走到了另一头，也会被送回来。

贝西在迷宫里迷路了，她希望走到迷宫的终点，终点以  $=$  标示。她确定出口一定在迷宫的边界上，除了出口所在的方格外，迷宫的四条边界上都是玉米。请你帮助她用最少的移动步数来逃出迷宫吧，假设逃脱的路线一定是存在的，注意传送不算入贝西的移动次数。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $3 \leq N, M \leq 300$
- 第二行到第  $N+1$  行：第  $i+1$  行有  $M$  个字符，描述了迷宫中位于第  $i$  行方格的信息。其中
  - “#” 代表不能通行的玉米地
  - “.” 代表可以通行的草地
  - “@” 代表贝西的起始位置
  - “=” 代表出口
  - 大写字母“A”到“Z”总是两两出现的，代表一对传送点

### 输出格式

- 单个整数：表示贝西走出迷宫的最少步数

### 样例输入

```
5 6
####=##
#.W.##
#.####
#.@W##
#####
```

### 样例输出

```
3
```

### 解释

从起点向右走一步，达到 W 传送点，被传到另一端，再向右走一步，向上走一步，就走到了终点

## 来源

Corn Maze, 2011 Open

## 代号

cornmaze



## 70 ★★☆☆☆ 奶牛攀岩

攀岩是很有意思的活动。贝西想借助你的智慧来规划她的路线。

贝西所挑战的岩体呈长方形，高  $H$ 。岩体上有  $N$  块岩石可供攀爬中搭把手，第  $i$  块岩石的坐标为  $(X_i, Y_i)$ ，没有岩石的坐标是相同的。每块岩石之间的距离不小于 300。

贝西的蹄子很长，只要两块岩石间的距离不超过 1000，她就可以在这两块岩石间自由移动。一旦她爬到某块高度距离大于或等于  $H - 1000$  的岩石，那么下一步她就可以直接翻过岩体，获得奖励了。

一开始，贝西在地面上，她可以自由走动，爬到任何一块距离地面不超过 1000 的岩石上。假设一定存在一条从地面爬到墙顶的路线，请问贝西最少要碰几块岩石才能翻过岩体？

### 输入格式

- 第一行：两个整数  $H$  和  $N$ ， $1001 \leq H \leq 30000$ ， $1 \leq N \leq 10000$   $1 \leq N \leq 5000$ ， $1 \leq SX, SY \leq 1000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $X_i$  和  $Y_i$ ， $1 \leq X_i \leq 30000$ ， $1 \leq Y_i \leq H$ ，表示一块岩石的坐标

### 输出格式

- 单个整数：表示最少需要碰几块岩石才能翻过岩体

### 样例输入

```
3000 5
600 800
1600 1800
100 1300
300 2100
1600 2300
```

### 样例输出

```
3
```

### 解释

行进路线为  $(600, 800) \rightarrow (100, 1300) \rightarrow (300, 2100)$

来源

The Climbing Wall, 2006 Open

代号

wall

## 71 ★★★★★ 黄金莲花池

约翰有一个长方形池塘，可以分成  $N \times M$  个方格，其中一些是坚固得令人惊讶的莲花，还有一些是岩石，其余的只是纯净湛蓝的水。贝西正在池塘里练习芭蕾舞，想从起点跳到终点，她的舞步很像象棋中的马：每次跳跃可以横向移动两个 2 格加纵向移 1 格，纵向移动两个 2 格加横向移 1 格，中途必须跳在莲花上，不能呆在水里或岩石上。约翰发现贝西跳不到终点，因为池塘里缺了些必要的莲花。约翰计划帮助贝西，他可以在有水的格子加上莲花，但石头上不行，一贯吝啬的他只想添加最少数量的莲花。请帮助约翰确定至少要添加几朵莲花，顺便帮助他数一下放置这些莲花有多少种不同的方法。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N, M \leq 30$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有  $M$  个用空格分开的整数，描述了池塘里第  $i$  行的状态：0 为水，1 为莲花，2 为岩石，3 为起点，4 为终点

### 输出格式

- 第一行：单个整数：表示需要最少增加的莲花数，如果无解则输出 -1
- 第二行：单个整数，表示放置莲花的方案总数，保证这个数字小于  $2^{63}$ ，如果前一行是 -1，这行没有内容

### 样例输入

```
4 5
1 0 0 0 0
3 0 0 0 0
0 0 2 0 0
0 0 0 4 0
```

### 样例输出

```
2
3
```

### 解释

X 为莲花位置：

```
1 0 0 0 0    1 0 X 0 0    1 0 X 0 0
3 0 X 0 0    3 0 0 0 0    3 0 0 0 X
0 0 2 0 0    0 X 2 0 0    0 0 2 0 0
0 X 0 4 0    0 0 0 4 0    0 0 0 4 0
```

## 来源

Lilypad Pond, 2007 Feb

## 代号

lilypad

## Chapter 25

# Bellman-Ford

## 72 ★★☆☆☆☆ 虫洞

约翰在他牧场里发现了虫洞！虫洞是非常奇妙的物理现象，它是一种特殊的道路，沿着它前进可以从一个地点到达另一个地点，而且时间会发生倒流。

约翰的牧场有  $N$  个地点， $M$  条普通道路和  $W$  条虫洞。所有的普通道路都是双向通行的，其中第  $j$  条道路连接了第  $A_j$  个地点和第  $B_j$  个地点，通行会消耗  $D_j$  分钟；所有虫洞都是单向通行的，行走会使时间倒流。其中第  $i$  条虫洞从第  $S_i$  个地点出发，通向第  $T_i$  个地点，通行会使时间返回  $B_i$  分钟。

既然发现了虫洞，那么当然要设法回到过去，和过去的自己见面。约翰打算以第一个地点为起点，寻找可以让他返回起点，而且时间回到过去的路线。如果存在这样的路线，输出“YES”，否则输出“NO”。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $W$ ， $1 \leq N \leq 500, 1 \leq M \leq 2500, 1 \leq W \leq 200$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $A_i$ ， $B_i$  和  $D_i$ ， $1 \leq A_i, B_i \leq N, 1 \leq D_i \leq 10000$
- 第  $M+2$  行到第  $M+W+1$  行：第  $i+M+1$  行有三个整数  $S_i$ ， $T_i$  和  $B_i$ ， $1 \leq S_i, T_i \leq N, 1 \leq B_i \leq 10000$

### 输出格式

- 单独一行：如果存在，输出“YES”，否则输出“NO”

### 样例输入

```
3 2 1
1 2 3
2 3 4
3 1 8
```

### 样例输出

```
YES
```

### 解释

回到过去的路线为  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$

来源

Wormholes, 2006 Dec

代号

wormhole

解答

## 73 ★★☆☆☆ 奶牛观光

作为对奶牛辛勤工作的回报，约翰决定带她们去附近的大城市玩一天。这个城市有  $N$  个景点，参观第  $i$  个景点会给奶牛带来  $F_i$  点欢乐度。景点间有  $M$  条道路，道路都是单行道，第  $i$  条道路从  $S_i$  开始通向  $T_i$ ，长度为  $L_i$ 。

约翰让奶牛们自由活动一天，开始旅游的那天早晨，奶牛们可以选择从任意一个景点出发，约翰会开车把她们送到那里，在晚上结束的时候，奶牛必须回到这个起点和约翰汇合。

奶牛们想让欢乐度尽量大，但经讨厌走路，所以需要设计一条游览线路。定义一条游览线路的“欢乐指数”为该线路上所有景点的欢乐度之和与路程长度的比值。欢乐指数越大的线路越受奶牛的欢迎。在旅游的时候，奶牛们是乐意重复路过同一个景点的，但参观同一景点只记一次欢乐度。奶牛至少要参观两个景点，保证给定的道路系统里至少能找出一条环形线路，请帮助奶牛规划处一条欢乐指数最大的线路来吧。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $2 \leq N \leq 1000, 2 \leq M \leq 5000$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $F_i$ ， $1 \leq F_i \leq 1000$
- 第  $N+2$  行到第  $N+M+1$  行：第  $i+N+1$  行有三个整数  $S_i$ ， $T_i$  和  $L_i$ ， $1 \leq S_i, T_i \leq N, 1 \leq L_i \leq 1000$

### 输出格式

- 单个浮点数：表示最大的欢乐指数，以四舍五入法保留两位小数

### 样例输入

```
5 7
30
10
10
5
10
1 2 3
2 3 2
3 4 5
3 5 2
4 5 5
5 1 3
5 2 2
```

### 样例输出

```
5 2 2
```

### 解释

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 1$ ，欢乐度为 60，路长为 10



来源

Sightseeing Cows, 2007 Dec

代号

sightsee

解答

## 74 ★★★☆☆ 奶牛的站位

有  $N$  头奶牛正在排队，它们的编号为 1 到  $N$ ，约翰要给它们安排合适的排队位置，满足以下条件：

- 首先，所有奶牛要站在一条直线上。由于是排队，所以编号小的奶牛要靠前，不能让编号大的奶牛插队。但同一个位置可以容纳多头奶牛，这是因为它们非常苗条的缘故
- 奶牛喜欢和朋友靠得近点。朋友关系有  $F$  对，其中第  $A_i$  头奶牛和第  $B_i$  头奶牛是第  $i$  对朋友，它们的距离不能超过  $C_i$
- 奶牛还要和讨厌的同类保持距离。敌对关系有  $E$  对，其中第  $X_i$  头奶牛和第  $Y_i$  头奶牛是第  $i$  对敌人，它们的距离不能少于  $Z_i$

你能否帮助约翰找到一个合理的站位方法，满足所有奶牛的要求，而且让 1 号奶牛和  $N$  号奶牛间的距离尽量大？

### 输入格式

- 第一行：三个整数  $N$ ， $F$  和  $E$ ， $2 \leq N \leq 1000$ ,  $1 \leq F, E \leq 10000$
- 第二行到第  $F+1$  行：第  $i+1$  行有三个整数  $A_i$ ， $B_i$  和  $C_i$ ， $1 \leq A_i, B_i \leq N$ ,  $1 \leq C_i \leq 10^6$
- 第  $F+2$  行到第  $F+E+1$  行：第  $i+F+1$  行有三个整数  $X_i$ ， $Y_i$  和  $Z_i$ ， $1 \leq X_i, Y_i \leq N$ ,  $1 \leq Z_i \leq 10^6$

### 输出格式

- 单个整数：如果不存在满足所有条件的安排，输出  $-1$ ；如果 1 号奶牛和  $N$  号奶牛的距离可以任意大，输出  $-2$ ；否则输出 1 号奶牛和  $N$  号奶牛的最大距离

### 样例输入

```
4 2 1
1 3 10
2 4 20
2 3 3
```

### 样例输出

```
27
```

### 解释

一号奶牛站在坐标 0，二号奶牛站在坐标 7，  
三号奶牛站在坐标 10，四号奶牛站在坐标 27

## 来源

Layout, 2005 Dec

## 代码

layout

## Chapter 26

# Dijkstra

## 75 ★★★☆☆ 领取巧克力

情人节到了，奶牛们会收到约翰发来的巧克力。约翰的牧场有  $N$  块草地，这些草地的编号为 1 到  $N$ ， $M$  条双向通行的道路连接这些草地，其中第  $i$  条道路的长度为  $L_i$ ，连接的草地编号为  $S_i$  和  $T_i$ ，同一对草地间可能有多条道路。

牧场里有  $B$  头大牛，每头大牛都有一个对象，第  $i$  头大牛目前在草地  $P_i$  上，他喜欢的母牛在草地  $Q_i$  上。约翰在 1 号草地，为了拿到巧克力，大牛们先要走到 1，然后从 1 走到对象所在的地方。保证所有的草地之间都通过道路连通，请帮助这些大牛算一下，他们最短走多少路才能遇上自己的对象。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $B$ ， $1 \leq N \leq 50000, 1 \leq M \leq 100000, 1 \leq B \leq 25000$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $S_i$ ， $T_i$  和  $L_i$ ， $1 \leq S_i, T_i \leq N, 1 \leq L_i \leq 2000$
- 第  $M+2$  行到第  $M+B+1$  行：第  $i+M+1$  行有两个整数  $P_i$  和  $Q_i$ ， $1 \leq P_i, Q_i \leq N$

### 输出格式

- 第一行到第  $B$  行：第  $i$  行有一个整数，表示第  $i$  头大牛行走的最短距离

### 样例输入

```
6 7 3
1 2 3
5 4 3
3 1 1
6 1 9
3 4 2
1 4 4
3 2 2
2 4
5 1
3 6
```

### 样例输出

```
6
6
10
```

### 解释

第一头大牛从 2 走到 1，再经过 3，最终到 4；第二头大牛从 5 到 4，再走到 3，最后停在 1；第三头大牛从 3 走到 1，再走到 6

## 来源

Chocolate Giving, 2010 Feb

## 代码

cgiving

## 解答

## 76 ★★★☆☆ 次短路径

贝西所在的牧场一共有  $N$  个地点。 $M$  条双向通行的道路连接这些地点，其中第  $i$  条道路连接  $A_i$  和  $B_i$ ，长度为  $L_i$ 。贝西想从第一个地点走到第  $N$  个地点，由于路上风景不错，她决定不走最短路径，而选择次短路径。次短路径的长度严格大于最短路径。如果有两条路径的长度都是最短的，那么它们都不算次短路径。次短路径允许重复通过一些道路或地点。请你帮助贝西找出次短路径的长度吧，输入数据保证次短路径一定存在。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 5000$ ,  $1 \leq M \leq 10^5$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $A_i$ ， $B_i$  和  $L_i$ ， $1 \leq A_i, B_i \leq N$ ,  $1 \leq L_i \leq 5000$

### 输出格式

- 单个整数：表示从第一个点到最后一个点的次短路径长度

### 样例输入

```
4 4
1 2 100
2 4 200
2 3 250
3 4 100
```

### 样例输出

```
450
```

### 解释

最短路是  $1 \rightarrow 2 \rightarrow 4$ ，总长度为 300，次短路是  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ，总长度为 450

来源

Roadblocks, USACO 2006 Nov

代号

block

解答



## 77 ★★★★★☆ 道路翻新

约翰有  $N$  个牛棚，它们之间由  $M$  条双向道路连接，第  $i$  条道路连接第  $U_i$  个牛棚和第  $V_i$  个牛棚，通行这条道路需要  $T_i$  分钟。注意同一对牛棚之间可能有多条道路。

约翰每天要从第一个牛棚走到最后一个牛棚。为了节约花在路上的时间，他打算选择一些道路翻新。翻新后的道路可以非常快速地通过，通行时间视作 0。约翰最多能翻新  $K$  条道路，请帮助约翰计算一下，选择翻新哪些道路，才能使得从第一个牛棚走到第  $N$  个牛棚的通行时间之和变得最短。输入数据可以保证这两个牛棚是连通的。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $K$ ， $1 \leq N \leq 10000, 1 \leq M \leq 500000, 1 \leq K \leq 20$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $U_i$ ， $V_i$  和  $T_i$ ， $1 \leq U_i, V_i \leq N, 1 \leq T_i \leq 10^6$

### 输出格式

- 单个整数：表示道路翻新后的最短通行时间

### 样例输入

```
4 4 1
1 2 10
2 4 10
1 3 1
3 4 100
```

### 样例输出

```
1
```

### 解释

翻新  $3 \rightarrow 4$ ，最短路径为  $1 \rightarrow 3 \rightarrow 4$

## 来源

Revamping Trails, 2009 Feb

## 代号

revamp

## 解答

## 78 ★★★★★ 公路和航线

约翰正为他的牛奶寻找新市场。新市场里有  $N$  个城市，编号为 1 到  $N$ 。城市间有  $R$  条公路和  $P$  条航线。公路都是双向通行的，第  $i$  条公路连接  $A_i$  城和  $B_i$  城，过路费为  $C_i$ 。而航线都是单程的，第  $i$  条航线从  $X_i$  城出发，飞往  $Y_i$  城，机票费为  $Z_i$ 。由于航空公司竞争激烈，坐飞机有很多机会获得补贴，所以  $Z_i$  有时竟可能是个负数！由于这种奇怪的现象存在，政府出面协调了交通规划，保证如果从  $X$  城到  $Y$  城有一条航线，那就不能从  $Y$  城通过任何公路或航线的组合返回  $X$  城，从而杜绝了套利的可能性。

约翰拥有世界上最给力的奶牛，所以这些城市的居民都渴望他家的牛奶。约翰已决定将新市场的配送中心设在  $S$  城，他想知道将牛奶从  $S$  城送到各个城镇的最少费用分别是多少。

### 输入格式

- 第一行：四个整数  $N$ ,  $R$ ,  $P$  和  $S$ ,  $1 \leq S \leq N \leq 25000, 1 \leq P, R \leq 50000$
- 第二行到第  $R+1$  行：第  $i+1$  行有三个整数  $A_i$ ,  $B_i$  和  $C_i$ ,  $1 \leq A_i, B_i \leq N, 0 \leq C_i \leq 10000$
- 第  $R+2$  行到第  $R+P+1$  行：第  $i+R+1$  行有三个整数  $X_i$ ,  $Y_i$  和  $Z_i$ ,  $1 \leq X_i, Y_i \leq N, -10000 \leq Z_i \leq 10000$

### 输出格式

- 第一行到第  $N$  行：第  $i$  行有一个整数，表示从  $S$  城到  $i$  城的最小开销，如果根本到不了  $i$ ，输出“NO PATH”

### 样例输入

```
6 3 3 4
1 2 5
3 4 5
5 6 10
3 5 -100
4 6 -100
1 3 -10
```

### 样例输出

```
NO PATH
NO PATH
5
0
-95
-100
```

来源

Roads and Planes, 2011 Jan

代号

roadplane

解答

## Chapter 27

### Floyd

## 79 ★★☆☆☆☆ 奶牛比赛

约翰家有  $N$  头奶牛正在参加编程比赛，比赛已经进行了  $M$  轮。在每轮比赛里，会有两头奶牛 PK，在第  $i$  轮比赛里，第  $A_i$  头奶牛和第  $B_i$  头进行比赛，最后  $A_i$  战胜了  $B_i$ 。假设这些比赛完全靠运气，也就是说，实力高的奶牛一定会在比赛中胜出（每头奶牛的实力都是不等的）。约翰想搞个奶牛实力排名榜，但因为比赛次数还不够多，所以一些奶牛的名次还不确定。请帮助他统计一下，以目前公布的成绩来看，多少奶牛的名次是可以确定的。保证公布的成绩中间不会有矛盾的结果出现。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 100, 1 \leq M \leq 4500$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数：表示有多少奶牛的实力名次是确定的

### 样例输入

```
5 5
4 3
4 2
3 2
1 2
2 5
```

### 样例输出

```
2
```

### 解释

第二头奶牛一定是第四名，第五头奶牛一定是最后一名，其余三头的名次不定

来源

Cow Contest, 2008 Jan

代号

contest

解答

## 80 ★★☆☆☆☆ 奶牛跨栏

约翰的奶牛正在准备跨栏比赛。对奶牛来说，连续跳过几个较矮的栅栏是容易的，但一个比较高的栅栏却很难对付。所以，奶牛关心的是比赛路线里最高的一个栅栏。

奶牛的训练场中有  $N$  个地点，它们由  $M$  条跑道连接，第  $i$  条跑道的起点是第  $S_i$  个地点，终点是第  $E_i$  个地点。每条跑道都是单向的，在第  $i$  条跑道上存在一个高度为  $H_i$  的栅栏。

奶牛们要完成  $Q$  个任务。第  $i$  个任务要求奶牛从第  $A_i$  个地点跑到第  $B_i$  个地点，至于如何选择跑道，由奶牛自由决定。请你帮助奶牛为每个任务找到一条路线，使得这些路线上最高的栅栏尽量低一点。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $Q$ ， $1 \leq N \leq 300, 1 \leq M \leq 25000, 1 \leq Q \leq 40000$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $S_i$ ， $E_i$  和  $H_i$ ， $1 \leq S_i, E_i \leq N, 1 \leq H_i \leq 1000000$
- 第  $M+2$  行到第  $M+Q+1$  行：第  $i+M+1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 第一行到第  $T$  行：第  $i$  行有一个整数，表示从第  $A_i$  个地点跑到第  $B_i$  个地点的所有路线中，栅栏最高高度的最小值，如果一条路线都不存在，输出  $-1$

### 样例输入

```
5 6 3
1 2 12
3 2 8
1 3 5
2 5 3
3 4 4
2 4 8
3 4
1 2
5 1
```

### 样例输出

```
4
8
-1
```

### 解释

第一个任务：直接从 3 到 4 即可；第二个任务：先到 3，再到 2 为好



来源

Cow Hurdles, 2007 Nov

代号

hurdles

解答

## 81 ★★★★★☆ 过路费

翰家有  $N$  片草地，编号为 1 到  $N$ ，彼此之间由  $M$  条双向道路连接，第  $i$  条道路连接了  $A_i$  和  $B_i$ ，两片草地之间可能有多条道路，但没有道路会连接同一片草地，现有的道路可以保证任意两片草地都是连通的。

有一天，约翰宣布奶牛走路要收过路费，只要奶牛走过第  $i$  条道路，就要收费  $L_i$  元。此外，约翰还要求每头奶牛购买牌照，他为每片草地设置了牌照标准，如果奶牛购买的牌照价格低于某片草地的标准，她将被禁止进入那片草地。第  $i$  片草地的牌照标准为  $C_i$ 。

新政策一出，奶牛们敢怒不敢言，有  $Q$  头奶牛向你咨询最省钱的走路办法，第  $i$  头奶牛要从草地  $S_i$  走到  $T_i$ 。请你帮她们算算，选择什么样的路线才能最省钱？

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $Q$ ， $1 \leq N \leq 250, 1 \leq M \leq 10000, 1 \leq Q \leq 10000$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $C_i$ ， $1 \leq C_i \leq 10^6$
- 第  $N+2$  行到第  $N+M+1$  行：第  $i+1$  行有三个整数  $A_i$ ， $B_i$  和  $L_i$ ， $1 \leq A_i, B_i \leq N, 1 \leq L_i \leq 10^6$
- 第  $N+M+2$  行到第  $N+M+Q+1$  行：第  $i+N+M+1$  行有两个整数  $S_i$  和  $T_i$ ， $1 \leq S_i, T_i \leq N$

### 输出格式

- 第一行到第  $Q$  行：第  $i$  行有一个整数，表示从  $S_i$  到  $T_i$  的最低费用是多少

### 样例输入

```
5 7 2
2
5
3
3
4
1 2 3
1 3 2
2 5 3
5 3 1
5 4 1
2 4 3
3 4 4
1 4
2 3
```

### 样例输出

```
8
9
```

### 解释

最好办法分别是  $1 \rightarrow 3 \rightarrow 5 \rightarrow 4$  和  $2 \rightarrow 5 \rightarrow 3$

## 来源

Cow Toll Paths, 2009 Dec

## 代号

toll

## 解答

错误的想法：a 到 b 的最优路线依赖于 a 到 k 和 k 到 b 的。

问题：假设 a 到 k 比较近，购买高等级的牌照不合算，而 a 到 b 是一条比较长的路径，购买高等级的牌照是合算的，可见不能直接由子问题推得。

错误的想法 2：将点上的费用分解到边上去。这是不行的。因为点上的聚合操作是  $\max$ ，而边上的操作是取和，操作的方式不同，不可能相容。

正确但不快的做法：先枚举购买什么样的牌照，然后再求最短路，复杂度  $N^4$

## 82 ★★★★★ 奶牛接力

为增强奶牛的体质，约翰举办了一场接力跑比赛。有  $K$  头奶牛报名参加比赛，你需要为它们设计一条接力路线。

比赛现场有一些接力位置，每个位置用 1 到 1000 以内的整数编号。这些位置间有  $M$  条双向道路，第  $i$  条路的长度为  $L_i$ ，连接了  $U_i$  和  $V_i$ ，没有两条道路会重复连接同一对位置。由于奶牛数量较多，路线允许重复经过一些道路或位置，但路线的起点必须在  $S$ ，终点必须在  $E$ 。由于有  $K$  头奶牛参加比赛，所以需要寻找一条从  $S$  到  $E$ ，恰好经过  $K$  条道路的路线，这样每头奶牛都可以恰好跑一条路。没有奶牛愿意跑两段路或者不跑，所以约翰一定要选择一条恰好经过  $K$  条道路的路线。输入数据保证这样的路线一定是存在的，由于奶牛不擅长运动，所以请你帮助它们找到一条总长度最短的接力跑路线吧。

### 输入格式

- 第一行：四个整数  $K, M, S$  和  $E$ ， $2 \leq K \leq 10^6, 2 \leq M \leq 100, 1 \leq S, E \leq 1000$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $L_i, U_i$  和  $V_i$ ， $1 \leq L_i \leq 1000, 1 \leq U_i, V_i \leq 1000$

### 输出格式

- 单个整数：表示起点为  $S$ ，终点为  $E$ ，且恰好经过  $K$  段路的最短路线长度

### 样例输入

```
2 6 6 4
11 4 6
4 4 8
8 4 9
6 6 8
2 6 9
3 8 9
```

### 样例输出

10

### 解释

最短路线为  $6 \rightarrow 8 \rightarrow 4$ ，或  $6 \rightarrow 9 \rightarrow 4$

## 来源

Cow Toll Paths, 2009 Dec

## 代号

relays

## 解答

最慢的做法。改进到二分。最后用矩阵，定义矩阵的新运算

## Chapter 28

# 深度优先遍历

## 83 ★★★★★☆ 明星奶牛

每头奶牛都梦想成为牛棚里的明星。被所有奶牛喜欢的奶牛就是一头明星奶牛。所有奶牛都是自恋狂，所以每头奶牛总是喜欢自己的。奶牛之间的“喜欢”是可以传递的——如果  $A$  喜欢  $B$ ， $B$  喜欢  $C$ ，那么  $A$  自然也会喜欢  $C$ 。牛栏里共有  $N$  头奶牛，已知有  $M$  条信息，第  $i$  条信息说明第  $A_i$  头奶牛喜欢第  $B_i$  头奶牛，请你根据这些信息统计一下，有多少头奶牛是明星奶牛。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 1000$ ,  $1 \leq M \leq 50000$
- 第二行到第  $M+1$  行：第  $i+1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数：表示明星奶牛的数量

### 样例输入

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

### 样例输出

```
1
```

### 解释

只有第三头奶牛是明星奶牛

来源

Popular Cows, 2003 Fall

代号

popular

解答



## 84 ★★★★★☆ 冗余路径

约翰有  $N$  个牧场。  $M$  条双向道路连接这些牧场，其中第  $i$  条道路连接第  $A_i$  个牧场和第  $B_i$  个牧场。目前所有牧场之间都是连通的。

奶牛们希望使得每对牧场之间至少存在两条不重合的路径。所谓不重合，是指这两条路径不会经过同一条道路，经过同一个牧场是没关系的。两条连接同一对牧场的道路算做不同的路。请帮助约翰增加一些道路，满足奶牛的要求，而且使得新建的道路最少。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 5000$ ,  $1 \leq M \leq 10000$
- 第二行到第  $M + 1$  行：第  $i + 1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数：表示必须增加的最少道路条数

### 样例输入

```
7 7
1 2
2 3
3 4
2 5
4 5
5 6
5 7
```

### 样例输出

```
2
```

### 解释

可以连接  $(1, 6)$  和  $(4, 7)$

来源

Redundant Paths, 2006 Jan

代号

rpaths

解答

## Chapter 29

# 匹配

## 85 ★★★☆☆ 小行星群

贝西驾驶着宇宙飞船，准备穿过危险的小行星群。它所在的星域可以分为  $N \times N$  个网格，小行星存在于其中的  $M$  个网格。贝西要安全通过这片星域，必须消灭所有的小行星。好在她有一件终极兵器——地图炮。只要一发炮弹就可以清除同一行或同一列网格里的所有小行星。这种弹药很贵，所以贝西希望尽量少来几发。给出所有的小行星的位置，请算出最少需要几发炮弹才能消灭它们。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 500, 1 \leq M \leq 10^4$
- 第二行到第  $M+1$  行：第  $i+1$  行有两个整数  $R_i$  和  $C_i$ ， $1 \leq R_i, C_i \leq N$

### 输出格式

- 单个整数：表示最少需要发射的多少炮弹

### 样例输入

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

### 样例输出

```
2
```

### 解释

向第一行开炮，消灭 (1,1) 和 (1,3)，再向第二列开炮，消灭 (2,2) 和 (3,2)

来源

Asteroids, 2005 Nov

代号

asteroid

解答

## 86 ★★★★★☆ 泥泞的牧场

约翰的牧场下起了雨，雨水对牧草的生长有好处，但却使得没有长草的土地变得泥泞。牧场可分为  $R \times C$  个方格，其中一些是草地，另一些已经变成了泥地。奶牛们是很爱干净的动物，它们不想在吃草时把蹄子弄脏。于是约翰打算用些木板把所有泥地盖住。每块木板呈长方形，宽度恰好够盖住一个方格，长度可以由约翰任意选择。几块木板可以重叠覆盖一块泥地，但约翰不希望木板盖住任何一块草地。请问约翰最少需要多少木板才能覆盖所有的泥地？

### 输入格式

- 第一行：两个整数  $R$  和  $C$ ， $1 \leq R, C \leq 50$
- 第二行到第  $R+1$  行：第  $i+1$  行有  $C$  个字符串，表示第  $R$  的地形，“\*”代表泥地，“.”代表草地

### 输出格式

- 单个整数：表示最少需要多少木板

### 样例输入

```
4 4
*.*.
.***
***.
..*.
```

### 样例输出

```
4
```

来源

Muddy Fields, 2005 Jan

代号

cover

解答

## Chapter 30

# 网络流



## 87 ★★★☆☆ 奶牛晚宴

奶牛是非常挑剔的吃货，每头牛都有自己喜欢的食物和饮料。约翰做了  $F$  种食物， $D$  种饮料，分别以 1 到  $F$  和 1 到  $D$  编号，每种主食和饮料都只够一头牛吃的量。现在有  $N$  头奶牛饿得哞哞叫，但它们不会吃自己不喜欢的东西。要喂饱一头牛，就必须提供一份它喜欢的食物和饮料。请问约翰应该怎样分配各种食物和饮料，才能喂饱尽可能多的奶牛？

### 输入格式

- 第一行：三个整数  $N$ ， $F$  和  $D$ ， $1 \leq N \leq 100, 1 \leq F \leq 100, 1 \leq D \leq 100$
- 第二行到第  $N + 1$  行：第  $i + 1$  行首先由两个整数  $F_i$  和  $D_i$ ， $1 \leq F_i \leq F, 1 \leq D_i \leq D$ ，接下来有  $F_i$  个整数，表示第  $i$  头牛喜爱的食物编号，有  $D_i$  个整数，表示第  $i$  头牛喜爱的饮料编号

### 输出格式

- 单个整数：表示最多能喂饱多少奶牛

### 样例输入

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

### 样例输出

```
3
```

### 解释

第二头牛吃 2 号食品 and 2 号饮料，第三头牛吃 1 号食品 and 1 号饮料，第四头牛吃 3 号食品 and 3 号饮料

来源

Dining, 2007 Open

代号

dining

## 88 ★★★★★☆ 游览牧场

约翰家有  $N$  间牛棚， $M$  条双向道路连接了这些牛棚，第  $i$  条道路连接了第  $A_i$  间牛棚和第  $B_i$  间牛棚，长度为  $L_i$ 。所有牛棚中最好的是第一间和最后一间，所以当有朋友来访时，他会带着朋友从第一间牛棚走到第  $N$  间牛棚，然后再回到第一间牛棚。约翰想给朋友多看看乡村不同的景色，所以希望来回的路上不重复经过任何一条道路，不过重复经过一间牛棚是允许的。请帮助约翰选择一条路线，使得往返路径的总长度最短。输入数据保证路线总是存在的。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 1000$ ,  $1 \leq M \leq 10000$
- 第二行到第  $M+1$  行：第  $i+1$  行有三个整数  $A_i$ ,  $B_i$  和  $L_i$ ， $1 \leq A_i, B_i \leq N$ ,  $1 \leq L_i \leq 35000$

### 输出格式

- 单个整数：表示最短路线的总长度

### 样例输入

```
4 5
1 2 1
2 3 1
3 4 1
1 3 2
2 4 2
```

### 样例输出

```
6
```

### 解释

```
1 → 2 → 4 → 3 → 1
```

来源

Farm Tour, 2003 Feb

代号

tour

## 89 ★★★★★ 地震破坏

威斯康星发生了一场地震！约翰的牧场遭到了打击，有一些牛棚变成了废墟，如果一间牛棚遭到了破坏，那么所有和它相连的道路都不能使用了。

约翰有  $N$  个牛棚，编号为 1 到  $N$ ，有  $M$  条双向道路连接这些牛棚，第  $i$  条道路连接的牛棚是  $A_i$  和  $B_i$ ， $A_i$  可能等于  $B_i$ ，也可能有多条道路连接同一对牛棚。

约翰让奶牛们集中到 1 号牛棚避难。有  $P$  头奶牛通过手机向约翰求救，她们的遭遇类似：好消息是她们所在的牛棚没有损坏，坏消息是由于其他的牛棚遭到破坏，所以她们找不到任何可以走到 1 号牛棚的路径。假设发送报告的第  $i$  头奶牛目前在牛棚  $R_i$  里，那么以最乐观的情况来看，至少有多少牛棚被破坏了呢？

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $P$ ， $1 \leq P \leq N \leq 3000$ ， $1 \leq M \leq 20000$
- 第二行到第  $M+1$  行：第  $i+1$  行有两个整数  $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$
- 第  $M+2$  行到第  $M+P+1$  行：第  $i+M+1$  行有一个整数  $R_i$ ， $1 \leq R_i \leq N$ ，保证每个  $R_i$  都是不同的

### 输出格式

- 单个整数：表示至少有几个牛棚被破坏了

### 样例输入

```
5 5 2
1 2
2 3
3 5
2 4
4 5
4
5
```

### 样例输出

```
1
```

### 解释

最乐观的情况是只有 2 号牛棚坏了

来源

Earthquake Damage 2, 2009 Mar

代号

damage

## 90 ★★★★★ 玩具消毒

贝西的生日快到了，她准备举办一个持续  $N$  天的庆典。在庆典的第  $i$  天，她要为小伙伴们准备  $T_i$  份玩具。玩具可以从商店购买，每件  $P$  元。为了节约，在每天庆典结束之后，贝西会回收小伙伴们用过的玩具，选择送全部或部分玩具去消毒，经过消毒的玩具可以重复使用。

消毒店提供两种消毒方法，一种比较快，每件收费  $A$  元，隔  $S$  个夜晚能拿到；另一种比较慢，每件收费只要  $B$  元，但取货要隔  $T$  个夜晚。消毒后的玩具如果不立即使用，可以暂存起来。请问贝西应该怎么做才能花最少的钱为小伙伴们提供玩具？

### 输入格式

- 第一行：六个整数  $N, S, T, A, B$  和  $P$ ， $4 \leq N \leq 10^5, 1 \leq S \leq T \leq N, 1 \leq B \leq A \leq P \leq 60$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $T_i$ ， $1 \leq T_i \leq 50$

### 输出格式

- 单个整数：表示为了提供玩具所需要的最小总费用

### 样例输入

```
4 1 2 2 1 3
8
2
1
6
```

### 样例输出

```
35
```

### 解释

第一天买 8 份玩具，晚上送 2 件去快速消毒，6 件去慢速消毒；第二天送 1 件去慢速消毒，最后两天不花钱

## 来源

Toys, 2008 Nov

## 附注

原题中没有说明快洗速度一定快于慢洗，且数据中有这样的数据，删去这些平凡的数据。

## 代号

toy



## Chapter 31

### 动态规划（一）

## 91 ★★☆☆☆☆ 接住苹果

奶牛喜欢吃苹果。约翰有两棵苹果树，有  $N$  只苹果会从树上陆续落下。如果掉苹果的时候，贝西在那棵树下，她就能接住苹果。贝西一开始在第一棵树下。在苹果掉落之前，她有足够的时间来回走动，但她很懒，最多只愿意移动  $K$  次。请计算一下她最多可以接住几只苹果。

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq N \leq 1000, 1 \leq K \leq 30$
- 第  $i+1$  行有一个整数  $T_i$ ，表示第  $i$  只苹果从哪棵树上掉落，1 表示从第一棵树，2 表示从第二棵树

### 输出格式

- 单个整数：表示能接住的最大苹果数量

### 样例输入

```
7 2
2
1
1
2
2
1
1
```

### 样例输出

```
6
```

### 解释

先待在第一棵树下接住两个，然后移动到第二棵树下接住两个，再返回第一棵树接住最后两个

来源

Apple Catching, 2004 Nov

代号

bcatch

解答

## 92 ★★☆☆☆☆ 方形牛棚

约翰有一片很大的农场，呈正方形，可均匀地分为  $N \times N$  个方格，一些方格是空地，另一些已经有建筑了。约翰打算从中选择一片没有建筑的空地，建造一座牛棚，牛棚必须是正方形的，面积越大越好。请你帮助约翰计算一下这座牛棚的边长吧。在下面这个例子中，X 代表不能建造牛棚的区域：

	1	2	3	4	5	6	7	8
1								
2		X				X		
3								
4								
5								
6			X					
7								
8								

最大的牛棚存在于右下角，大小为  $5 \times 5$ 。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 1000, 1 \leq M \leq 10000$
- 第二行到第  $M + 1$  行：每行有两个整数  $X$  和  $Y$ ，表示一个不能建造牛棚的方格坐标， $1 \leq X, Y \leq N$

### 输出格式

- 单个整数：表示最大牛棚的边长

### 样例输入

```
8 3
2 2
2 6
6 3
```

### 样例输出

```
5
```

来源

Big Barn, 1997 Fall

代号

bigbrn

解答

## 93 ★★☆☆☆ 滑雪课程

贝西去科罗拉多州去滑雪，不过她还不太会玩，只是个能力为 1 的渣渣。贝西从 0 时刻进入滑雪场，一到  $T$  时刻就必须离开。滑雪场里有  $N$  条斜坡，第  $i$  条斜坡滑行一次需要  $D_i$  分钟，要求游客的能力达到  $C_i$  或以上时才能进入。贝西决心参加一些滑雪课程以提高自己的素质，这样可以在有限的时间内多滑几次坡。

滑雪场提供了  $S$  门课程。第  $i$  门课的开始时刻为  $M_i$ ，持续  $L_i$  分钟，如果想参加课程，就不能迟到或早退。上完课之后，贝西的滑雪能力将变成  $A_i$ 。注意，不是能力增加  $A_i$ ，而是变成  $A_i$ ，所以乱上课的话反而会使能力下降。贝西可以随意安排她的时间：滑雪、上课，或美美地喝上一杯可可汁。请问她如何安排上课和滑雪的时间，滑坡的次数才能达到最大？

### 输入格式

- 第一行：三个整数  $T, S$  和  $N$ ， $1 \leq T \leq 10^4, 1 \leq S \leq 100, 1 \leq N \leq 10^5$
- 第二行到  $S+1$  行：第  $i+1$  行描述了第  $i$  门课程，分别为  $M_i, L_i$  和  $A_i$ ， $1 \leq M_i, L_i \leq 10^4, 1 \leq A_i \leq 100$
- 第  $S+2$  行到  $S+N+1$  行：第  $S+i+1$  行描述了第  $i$  条斜坡，分别为  $C_i$  和  $D_i$ ， $1 \leq C_i \leq 100, 1 \leq D_i \leq 10^4$

### 输出格式

- 单个整数，表示贝西可以滑完的最大次数

### 样例输入

```
10 1 2
3 2 5
4 1
1 3
```

### 样例输出

```
6
```

### 解释

先滑 1 次二号斜坡，然后去上课，再去一号斜坡连滑 5 次

## 来源

Ski Lessons, 2009 Open

## 代号

ski

## 解答

效率低的 dp。记时间和能力

高效的 dp

## 94 ★★★☆☆ 滑雪比赛

贝西参加了一场高山急速滑雪比赛，滑道总长度为  $L$ 。出发时，她的初速度为 1，贝西可以加速或减速，每过 1 米，她能将速度增加 1、减少 1 或保持不变。在滑雪的过程中，贝西会遇到  $N$  个转弯点，编号为  $i$  的转弯点距离起点有  $T_i$  米。安全起见，贝西到达  $i$  号转弯点时的速度不能超过  $S_i$ 。穿过终点的速度是没有限制的。请问在整个比赛过程中，贝西能够达到的最大速度是多少？

### 输入格式

- 第一行：两个整数  $L$  和  $N$ ， $2 \leq L \leq 10^9, 1 \leq N \leq 10^5$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $T_i$  和  $S_i$ ， $1 \leq T_i < L, 1 \leq S_i \leq 10^9$

### 输出格式

- 单个整数：表示贝西在比赛过程中能够达到的最大速度

### 样例输入

```
14 3
7 3
11 1
13 8
```

### 样例输出

```
5
```

### 解释

第一次达到最高速度的位置在距离起点 4 米处



来源

Bobsledding, 2009 Dec

代号

bobsled

解答

## Chapter 32

## 动态规划（二）

## 95 ★★☆☆☆☆ 奶牛飞盘队

老唐最近迷上了飞盘，约翰想和他一起玩，于是打算从他家的  $N$  头奶牛中选出一些组成一支队伍。每只奶牛的能力为整数，第  $i$  头奶牛的能力为  $A_i$ ，一支队伍的总能力就是所有队员能力的总和。

约翰的幸运数字是  $M$ ，他要求队伍的总能力必须是  $M$  的倍数。你能帮他算一下符合要求的组合有多少种吗？由于这个数字很大，只要输出答案除以  $10^8$  的余数就可以了。当然队伍里至少要有一头奶牛。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N \leq 2000$ ,  $1 \leq M \leq 1000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $1 \leq A_i \leq 10^5$

### 输出格式

- 单个整数：表示方案总数除以  $10^8$  的余数

### 样例输入

```
4 5
1
2
8
2
```

### 样例输出

```
3
```

### 解释

因为有两个 2，所以有两个  $8 + 2$ ，第三种方案是  $2 + 2 + 1$

## 来源

Cow Frisbee Team, 2009 Mar

## 代号

fristeam

## 96 ★★☆☆☆ 滑雪缆车

科罗拉多州的山脉是二维平面上的一条折线。这条折线由  $N$  个端点， $N - 1$  段线段组成，第  $i$  个端点的横坐标就是  $i$ ，纵坐标是  $H_i$ ，纵坐标代表高度，也可以称为海拔。

罗恩打算为奶牛建造一个滑雪场，为此要在山脉上规划一条缆车线路。缆线也是一条折线，由若干段缆绳组成，起点在山脉的第一个端点，终点在最后一个端点。每段缆绳可以贴着山脉的轮廓，也可以悬浮于空中，跳过山脉上几个海拔低的端点。每段缆绳的水平跨度有限制，不能超过给定的整数  $K$ 。罗恩需要在每段缆绳的端点处修建支柱，用来固定缆绳。请帮助他规划一下，选择在山脉的哪些端点上修建，才能使得支柱数量最少？注意，根据题意，起点和终点上是一定要修建的。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $2 \leq N \leq 5000$ ， $1 \leq M \leq N - 1$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $H_i$ ， $0 \leq H_i \leq 10^9$

### 输出格式

- 单个整数：表示最少需要修建的支柱数量

### 样例输入

```
13 4
0
1
0
2
4
6
8
6
8
8
9
11
12
```

### 样例输出

```
5
```

### 解释

最优方案是把支柱设在 1, 5, 7, 9, 13。5 不能直接连 9，因为 9 的海拔较高，1 不能直接连 7，因为跨度超过了  $K$ 。

## 来源

Ski Lift, 2006 Mar

## 代号

skilift

## 97 ★★★☆☆ 奶牛自行车队

$N$  头奶牛组队参加自行车赛。车队在比赛时排成一列，需要绕场  $S$  圈。由于空气阻力的作用，领队奶牛消耗的体力要比后面的多。每头奶牛的初始体力都是相同的，记作  $M$ ，体力减为负数的奶牛只能中途退赛，体力也不会比赛途中恢复，但最后只要有一位队员到达终点就算完成了比赛。

比赛最小的单位时间是分钟。车队在每分钟必须绕赛场整数圈，最少是每分钟一圈。如果车队在一分钟里绕场  $x$  圈，领队奶牛会消耗  $x^2$  点体力，跟在后面的所有奶牛将会消耗  $x$  点体力。每分钟开始的时刻，车队可以自由选择是否换下领队奶牛，让其他奶牛做领队，并且设定这一分钟的速度。如果设定该分钟的速度为  $x$  圈，则要保证领队奶牛的体力至少要大于  $x^2$ 。

作为它们教练，请你计划一下，应采用什么样的策略才能让车队以最快的时间完成比赛？输入数据保证  $S \leq M$ ，因此一定存在完成比赛的方案。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $S$ ， $1 \leq N \leq 20, 1 \leq S \leq M \leq 100$

### 输出格式

- 单个整数：表示最早完成比赛的时间

### 样例输入

3 30 20

### 样例输出

7

### 解释

时 间	领 队	速 度	A 的体力	B 的体力	C 的体力	完成圈数
1	A	5	5	25	25	5
2	A	2	1	23	23	7
3	B	4	-	7	19	11
4	B	2	-	3	17	13
5	B	3	-	0	8	16
6	C	2	-	-	4	18
7	C	2	-	-	0	20

来源

Cow Cycling, 2002 Feb

代号

cycling



## 98 ★★★★★☆ 打扫食槽

牧场里有  $N$  头奶牛，约翰向它们提供  $M$  种食物，第  $i$  头奶牛吃的是第  $A_i$  种食物。约翰每天都要打扫食槽，这活很累。奶牛沿着食槽排成一条直线，约翰在打扫时，可以将食槽分割成几个区间，如果一段区间中有  $K$  种不同的食物，那么打扫这段区间的时间就是  $K^2$ 。请问约翰应该怎么划分区间才能使打扫整个食槽的时间之和最少。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq M \leq N \leq 40000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $1 \leq A_i \leq M$

### 输出格式

- 单个整数：表示约翰完成打扫的最短时间

### 样例输入

13 4

1

2

1

3

2

2

3

4

3

4

3

1

4

### 样例输出

11

### 解释

前四头各成一段，第五段两个 2，第六段为 3, 4, 3, 4, 3，最后两头各成一段， $1 \times 4 + 1 + 4 + 1 \times 2 = 11$

## 来源

Cleaning Up, 2009 Mar

## 代号

cleanup

99 ★★★★★☆ 连击游戏

贝西正在打格斗游戏。游戏里只有三个按键，分别是“A”、“B”和“C”。游戏中有  $N$  种连击模式，第  $i$  种连击模式以字符串  $S_i$  表示，只要贝西的按键中出现了这个字符串，就算触发了一次连击模式。不同的连击模式是独立计算的，如果几个连击模式同时出现在贝西的按键顺序里，就算有重叠部分，也可以同时算作触发了多个模式。

假如有三个连击模式，分别是“AB”，“BA”，“ABC”，而贝西按下了“ABABC”，那么她一共触发了四次连击。假设贝西一共可以按  $K$  次键，那么她最多能触发多少次连击呢？

输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq N \leq 20, 1 \leq K \leq 1000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个字符串  $S_i$ ， $S_i$  中由大写字符“A”、“B”、“C”构成，每个字符串长度在 1 到 15 之间

输出格式

- 单个整数：表示贝西可以完成的最多连击数量

样例输入

3 7  
ABA  
CB  
ABACB

样例输出

4

解释

贝西按下 ABACBCB，可以获得四次连击

来源

Video Game Combos, 2012 Jan

代号

combos

解答

## Chapter 33

# 动态规划（三）——区间型

## 100 ★★☆☆☆ 金币游戏

贝西和她的朋友邦妮挖到了一个宝箱，里面藏着  $N$  枚金币！但金币对奶牛没用，她们一直用这些金币来玩游戏。游戏的目的是看谁拿走的金币最值钱。

游戏开始之前，金币被排成一行，在第  $i$  个位置的金币面值为  $C_i$ 。贝西和邦妮轮流取金币，每次她们只能从队头或队尾的两块金币中挑一块拿走，不能一块也不拿。所有金币取完之后，游戏就结束了。贝西和邦妮都是非常聪明的，会采用最好的策略让自己取到的金币面值之和最大。假设贝西是先手，请帮她计算一下，她最多能拿走多少钱的金币？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 5000$
- 第二行到  $N + 1$  行：第  $i + 1$  行有一个整数： $C_i$ ， $1 \leq C_i \leq 5000$

### 输出格式

- 单个整数：先手可以拿到的最大面值之和

### 样例输入

4  
30  
25  
10  
35

### 样例输出

60

### 解释

先手取 35，后手取 30，先手再取 25，后手最后取 10

## 来源

Treasure Chest, 2010 Dec

## 代号

treasure

## 分析

分析一下贪心策略能否解决本题。

第一种策略是每次直接选择面值最大的金币，这种策略捡了芝麻丢了西瓜，反例如下：

1, 1, 1, 1, 1, 1000, 2

在这个例子里，谁拿到了最大的数字 1000 就赢了，先手不该取 2，而是该耐心地等待后手被迫把 1000 露出来。

第二种策略更巧妙一点，错误也更加隐蔽一点。假设只考虑有偶数个金币的情况，分别算出奇数位置和偶数位置的金币面值之和，如果奇数位置的和比较大，则先手一直盯着选奇数位置，否则，一直选偶数位置。由于金币总是偶数个的，所以先手确实可以做到不论后手如何选择，总能取走所有奇数位置的金币，而把所有偶数位置的金币留给后手。

这个策略的问题在于，先手没必要把自己总是限制在奇数位置上，举个反例：

1000, 100, 10, 9, 99, 999

先手可以拿到前三个，这是因为对后手来说，取到 9 开头的数字比取完其余所有的数字更重要。

这两种策略其实各走了极端：前者失败在不顾对手选择，过早暴露了有价值的金币，后者的失败在于太在意限制对手，而失去了更有价值的金币。

否决贪心想法后，尝试从归纳的角度来解决问题。

## 解答

考虑贝西先手面对的局面为  $C_1, \dots, C_N$ ，她有两种选择：

- 取走  $C_1$ ，剩下  $C_2, \dots, C_N$ ，并且成为这个局面的后手
- 取走  $C_N$ ，剩下  $C_1, \dots, C_{N-1}$ ，并且成为这个局面的后手

可见贝西选择取  $C_1$  还是  $C_N$ ，关键看剩下的局面中后手能拿多少金币。乍看之下，后手的最优价值求起来很复杂，但我们有一个非常简单的公式：

先手取到的面值 + 后手取到的面值 = 所有金币的面值之和

设  $f[i, j]$  为当桌上只剩下  $C_i, C_{i+1}, \dots, C_j$  时，先手时所能取到的最佳面值。则根据上述分析

$$f[1, N] = \max \begin{cases} C_1 + (C_2 + C_3 + \dots + C_N - f[2, N]) \\ C_N + (C_1 + C_2 + \dots + C_{N-1} - f[1, N-1]) \end{cases}$$

令  $S_k = C_1 + \cdots + C_k, S_0 = 0$ , 对上式做个化简, 得到

$$f[1, N] = S_N - S_0 - \min\{f[2, N], f[1, N-1]\}$$

求  $f[1, N]$ , 要先求  $f[2, N]$  和  $f[1, N-1]$ , 要求  $f[2, N]$ , 就要先求  $f[3, N]$  和  $f[2, N-1]$ ……(此处应有一张图)

将  $f[i, j]$  看成一个区间, 每次归纳都会减少区间的长度, 最后区间长度会减少到 1, 当区间长度为 1 时, 自然有  $f[i, i] = C_i$ 。确定了边界之后, 我们就可以写出完整的方程来了:

$$\begin{aligned} f[i, j] &= S_j - S_{i-1} - \min\{f[i+1, j], f[i, j-1]\}, & 1 \leq i < j \leq N \\ f[i, i] &= C_i, & 1 \leq i \leq N \end{aligned}$$

## 实现

实现时要注意细节, 其中最重要的是弄清子问题之间的计算顺序。初学者可能会写出这样的代码来:

(此处应有一段伪代码)

这种写法是错误的。原因在于  $f[2, N]$  是  $f[1, N]$  的子问题, 应该先算出  $f[2, N]$  再算  $f[1, N]$ , 但在这段代码里, 先算完的竟然是  $f[1, N]$ , 此时  $f[2, N]$  甚至没有初始化, 当然不会得到正确的答案。

解决方法有三个, 第一种方案比较巧妙: 只要把  $i$  的循环方向改成倒序即可:

(此处应有一段伪代码)

第二种, 点长式

第三种, 记忆化搜索

(伪代码用 flag 标记算过没, cpp 代码用 `ans[i,j]` and `1 shl 30`)



## 101 ★★☆☆☆ 修改回文

追踪贝西是一件非常棘手的任务。为此约翰安装了一套自动系统，并且为贝西设置了电子标签，当她通过扫描器的时候，系统可以识别她的身份，从而锁定她的位置。

贝西的电子标签是一个长度为  $M$  的字符串，记做  $C_1C_2\ldots C_M$ ，其中每个字母都是小写字母，而且属于一个字母集  $\Sigma = \{S_1, S_2, \ldots, S_N\}$ 。

当然，贝西是非常顽皮的，她有时会故意在通过扫描器的时候倒着走。这样一来，扫描器会把字符串  $abcb$  读成  $bcba$ ，机器就会发生故障。约翰的解决办法很简单，不改造识别器，而是把标签改成回文。所谓回文就是一种倒序阅读等于顺序的字符串，比如  $abcba$ 。

约翰可以在电子标签的任意位置上插入或删除任意字符。假设要将  $abcb$  变成回文，可以在最后加个  $a$  变成  $abcba$ ；也可以删去  $a$  和  $c$  让  $abcb$  变成  $bb$ 。不过修改电子标签是要付费的，修改每种字母的费用是固定的。请帮助约翰计算一下，把标签修改成为回文的最小费用之和是多少。

### 输入格式

- 第一行：两个整数  $M$  和  $N$ ， $1 \leq M \leq 2000, 1 \leq N \leq 26$
- 第二行：连续  $M$  个字母： $C_1C_2\ldots C_M$
- 第三行到第  $N+2$  行：第  $i+2$  行首先有一个小写字母  $S_i$ ，其次分别是增加和删除一个  $S_i$  的费用  $A_i$  和  $D_i$ ， $0 \leq A_i, D_i \leq 10000$

### 输出格式

- 单个整数：表示修改成回文的最小费用之和

### 样例输入

```
3 4
abcb
a 1000 1100
b 350 700
c 200 800
```

### 样例输出

```
900
```

### 解释

变成  $bcbabcb$  的代价为  $350+200+350=900$

## 来源

Cheapest Palindrome, 2007 Open

## 代号

cheappal

## 解答

## 102 ★★☆☆☆ 奶牛吃草

贝西走在一条笔直的道路，处于道路上坐标为  $K$  的地方。她突然发现周围长着  $N$  棵非常新鲜的青草！其中第  $i$  棵青草位于道路上坐标为  $X_i$  的地方。贝西每秒钟可以沿着道路的方向向前或向后移动一个坐标单位的距离。只要移动到青草所在的地方，就可以一口吞掉青草，贝西的食速很快，吃草的时间可以不计。

作为吃货的贝西当然要吃光所有的青草。不过，青草太新鲜了，在被吞掉之前，暴露在道路上的每棵青草每秒都会损失一单位的口感。请你告诉贝西，该怎样来回跑动，才能在口感损失之和最小的情况下吃掉所有的青草。

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq N \leq 1000, 1 \leq K \leq 10^6$
- 第二行到  $N+1$  行：第  $i+1$  行有一个整数  $X_i$ ， $1 \leq X_i \leq 10^6$

### 输出格式

- 单个整数：表示最小损失的口感之和

### 样例输入

```
4 10
1
9
11
19
```

### 样例输出

```
44
```

### 解释

先跑到 9，然后跑到 11，再跑到 19，最后到 1

来源

Grazing on the Run, 2005 Nov

代号

ontherun

解答

## 103 ★★★★★☆ 提交作业

贝西在啾啾大学选修了  $C$  门课，她要把所有作业分别交给每门课的老师，然后去车站和同学们一起回家。每个老师在各自的办公室里，办公室要等他们下课后才开，第  $i$  门课的办公室将在  $T_i$  分钟后开放。

所有的办公室都在一条笔直的走廊上，这条走廊长  $H$  个单位，一开始贝西在走廊的尽头一侧，位于坐标为 0 的地方。第  $i$  门课的办公室坐标位于坐标为  $X_i$  的地方，车站的坐标为  $B$ 。

贝西可在走廊上自由行走，每分钟可以向右或者向左移动一个单位，也可以选择停着不移动。如果走到一间已经开门的办公室，贝西就可以把相应的作业交掉了，走进办公室交作业是不计时间的。

请帮助贝西计算一下，从她开始交作业开始，直到到交完所有作业，再走到车站，最短需要多少时间。

### 输入格式

- 第一行：三个整数  $C$ ,  $H$  和  $B$ ,  $1 \leq C \leq 1000, 1 \leq H \leq 1000, 0 \leq B \leq H$
- 第二行到  $C+1$  行：第  $i+1$  行有两个整数  $X_i$  和  $T_i$ ,  $0 \leq X_i \leq H, 0 \leq T_i \leq 10000$

### 输出格式

- 单个整数，表示贝西交完作业后走到车站的最短时间

### 样例输入

```
4 10 3
8 9
4 21
3 16
8 12
```

### 样例输出

```
22
```

### 解释

走到坐标 8 处，第 9 分钟交一本作业，等到第 12 分钟时，交另一本作业。再走到坐标 4 处交作业，最后走到坐标 3 处，交最后一本作业，此地就是车站所在位置，共用时 22 分钟

来源

Turning in Homework, 2004 Open

代号

turnin

解答

## Chapter 34

# 动态规划（四）——背包问题

## 104 ★★☆☆☆☆ 平分子集

奶牛正在学习算术，今天的作业是抄写 1 到  $N$  之间的所有整数，然后找到尽量多的分组方法，把这些整数分成总和相等的两部分。当  $N = 3$  时，只有一种分法：一边是  $\{1, 2\}$ ，另一边是  $\{3\}$ 。当  $N = 7$  时，就有四种分法了：

$\{1, 6, 7\}, \{2, 3, 4, 5\}$   
 $\{2, 5, 7\}, \{1, 3, 4, 6\}$   
 $\{3, 4, 7\}, \{1, 2, 5, 6\}$   
 $\{3, 5, 6\}, \{1, 2, 4, 7\}$

请你帮奶牛计算一下，对于给定的  $N$ ，会有多少分法？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50$

### 输出格式

- 单个整数：表示方案数，如果不存在任何分法，输出 0

### 样例输入

7

### 样例输出

4



## 来源

Subset Sums, 1998 Spring

## 代号

subset

## 解答

## 参考程序

```
1  #include <iostream>                                1  {todo}
2  using namespace std;
3  const int maxn = 60;
4  long long f[maxn*maxn/2];
5  int main() {
6      int n;
7      cin >> n;
8      int s = (n + 1) * n / 2;
9      if (s % 2 != 0) {
10         cout << 0 << endl;
11     } else {
12         int t = s / 2;
13         f[0] = 1;
14         for (int k = 1; k < n; k++)
15             for (int x = t; x >= k; x--)
16                 f[x] += f[x-k];
17         cout << f[t] << endl;
18     }
19     return 0;
20 }
```

## 105 ★★☆☆☆ 太空电梯

奶牛们准备升空了！为了进入轨道，他们计划建造一座太空电梯。电梯需要由各种材料堆叠而成，每块材料可以自由地叠在其他材料上面，电梯的总高度就是材料的长度之和。

奶牛拥有  $N$  种材料，第  $i$  种材料的长度为  $H_i$ ，安全海拔为  $A_i$ ，这种规格的材料有  $C_i$  块。所谓安全海拔，就是指第  $i$  种型号的材料只能安放在海拔小于或等于  $A_i$  的空间里，这是因为宇宙射线可能危及材料的安全。假设奶牛从海拔为零的地面上开始建造太空电梯，请帮助它们用这些材料堆出尽可能高的太空电梯来吧。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 400$
- 第二行到第  $N+1$  行：第  $i+1$  行有三个整数： $H_i$ ， $A_i$  和  $C_i$ ， $1 \leq H_i \leq 100, 1 \leq A_i \leq 40000, 1 \leq C_i \leq 10$

### 输出格式

- 单个整数：表示太空电梯的最高高度

### 样例输入

```
3
7 40 3
5 23 8
2 52 6
```

### 样例输出

```
48
```

### 解释

自底向上：先取四块第二号材料，再取两块第一号材料，再取六块第三号材料

来源

Space Elevator, 2005 Mar

代号

elevator

解答

## 106 ★★★☆☆ 奶牛会展

奶牛想证明它们是聪明而风趣的。为此，贝西筹备了一个奶牛博览会，她已经对  $N$  头奶牛进行了面试，确定了每头奶牛的智商和情商。

贝西有权选择让哪些奶牛参加展览。由于负的智商或情商会造成负面效果，所以贝西不希望出展奶牛的智商之和小于零，或情商之和小于零。满足这两个条件下，她希望出展奶牛的智商与情商之和越大越好，请帮助贝西求出这个最大值。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 100$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数： $S_i$  和  $F_i$ ，表示第  $i$  头奶牛的智商和情商， $-1000 \leq S_i, F_i \leq 1000$

### 输出格式

- 单个整数：表示情商与智商和的最大值。贝西可以不让任何奶牛参加展览，如果这样做是最好的，输出 0

### 样例输入

```
5
-5 7
8 -6
6 -3
2 1
-8 -5
```

### 样例输出

```
8
```

### 解释

选择第一头，第三头，第四头奶牛，智商和为  $-5 + 6 + 2 = 3$ ，情商和为  $7 - 3 + 1 = 5$ 。再加入第二号奶牛可使总和提升到 10，不过由于情商和变成负的了，所以是不允许的

来源

Cow Exhibition, 2003 Fall

代号

smrtfun

解答

## 107 ★★★☆☆ 促销日

一年一度的年末促销终于到了！商店里有  $K$  种商品，每种商品的价格依次为 1 到  $K$  元，商品的供应是无限的。遇到难得的促销活动，约翰揣着  $N$  元钱来到商店，不把这些钱花完是不行的。请问约翰有几种把钱花完的方法？每种价格恰好等于  $N$  元的商品组合视作一种花钱的方法。

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq N \leq 1000$ ,  $1 \leq K \leq 100$

### 输出格式

- 单个整数：表示恰好等于  $N$  元的商品组合种数

### 样例输入

5 3

### 样例输出

5

### 解释

五种方案分别为  $1+1+1+1+1$ ； $1+1+1+2$ ；  
 $1+1+3$ ； $1+2+2$ ； $2+3$

来源

Dollar Dayz, 2006 Jan

代号

ddayz

解答

## 108 ★★★★★☆ 股票市场

尽管奶牛天生谨慎，它们仍然在住房抵押贷款市场中太受打击，现在它们准备在股市上碰碰运气。贝西有内部消息，她知道  $S$  只股票在今后  $D$  天内的价格。

假设在一开始，她筹集了  $M$  元钱，那么她该怎样操作才能赚到最多的钱呢？贝西在每天可以买卖多只股票，也可以多次买卖同一只股票，交易单位必须是整数，数量不限。举一个牛市的例子。假设贝西有 10 元本金，股票价格如下：

股票	今天的价格	明天的价格	后天的价格
A	10	15	15
B	13	11	20

最赚钱的做法是：今天买入 A 股 1 张，到明天把它卖掉并且买入 B 股 1 张，在后天卖掉 B 股，这样贝西就有 24 元了。

### 输入格式

- 第一行：三个整数  $S$ ， $D$  和  $M$ ， $2 \leq S \leq 50, 2 \leq D \leq 10, 1 \leq M \leq 200000$
- 第二行到第  $S+1$  行：第  $i+1$  行有  $D$  个整数： $P_{i,1}$  到  $P_{i,D}$ ，表示第  $i$  种股票在第一天到最后一天的售价，对所有  $1 \leq j \leq D, 1 \leq P_{i,j} \leq 1000$

### 输出格式

- 单个整数：表示奶牛可以获得的最大钱数，保证这个数不会超过 500000

### 样例输入

```
2 3 10
10 15 15
13 11 20
```

### 样例输出

```
24
```



来源

Stock Market, 2009 Feb

代号

stock

解答

## Chapter 35

### 动态规划（五）——背包问题续

## 109 ★★☆☆☆ 三角牧场

奶牛有  $N$  块木板，第  $i$  块木板的长度为  $L_i$ ，它们要用这些木板作为栅栏，围出一个三角形的牧场。所有的木板都必须用上，不得浪费。她们聘请你为设计师，请你帮助它们围出一个尽量大的三角形。

### 输入格式

- 第一行：单个整数  $N$ ， $3 \leq N \leq 40$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $L_i$ ， $1 \leq L_i \leq 40$

### 输出格式

- 单个整数：表示最大三角形面积的平方。如果用这些木板无法搭出任何三角形，输出  $-1$
- 注意：英文原题要求输出最大面积乘 100 后的整数，此处修改一下保证数据为整数。

### 样例输入

5  
1  
1  
3  
3  
4

### 样例输出

48  
(692)

### 解释

面积最大的是边长为 4 的等边三角形

来源

Triangular Pastures, 2002 Feb

代号

pasture

解答

## 110 ★★★★★☆ 录制唱片

贝西想从奶牛摇滚乐队的  $N$  首歌里挑出一些录制一套专辑。专辑分  $K$  张唱片，每张唱片可容纳总长为  $C$  分钟的歌曲。第  $i$  首歌的长度为  $T_i$  分钟，录制唱片时，唱片之间的歌曲需要保持原本的顺序。第一张唱片里的歌必须排在第二张唱片前面，之后的每张唱片里也是如此。同时，每首歌曲必须完整地放在一张唱片里，不然，就只能舍去不录了。请问贝西应该选哪些歌曲，才能让专辑里装下的歌曲尽量多？

### 输入格式

- 第一行：三个整数  $N$ ， $C$  和  $K$ ， $1 \leq N \leq 100, 1 \leq C \leq 1000, 1 \leq K \leq 1000$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $T_i$ ， $1 \leq T_i \leq 1000$

### 输出格式

- 单个整数：表示可以装进专辑的歌曲数目

### 样例输入

```
4 5 2
4
3
4
2
```

### 样例输出

```
3
```

### 解释

第一张唱片装第一首歌，第二张唱片装第二首和第四首歌

来源

Raucous Rockers, 1996 Qualifying Round

代号

rockers

解答

## 111 ★★★★★☆ 电子游戏

翰的奶牛玩游戏成瘾！本来约翰是想把她们拖去电击治疗的，但是他发现奶牛们在玩游戏后生产了更多的牛奶，也就支持它们了。

但是，奶牛在选择游戏平台上的分歧很大：有些奶牛想买 Xbox 360 来跑《光晕 3》；有的奶牛想要任天堂 Wii 来跑《明星大乱斗 X》；还有奶牛想要在 PlayStation 3 上玩《潜龙谍影 4》。约翰只有  $V$  元钱，不够多，要做一些取舍才行。

已知市面上有  $K$  种游戏平台，如果想玩第  $i$  种平台的游戏，必须先买一台该平台的游戏机，价格为  $C_i$ 。第  $i$  种平台上有  $S_i$  种游戏，其中第  $j$  个游戏的价格为  $P_{i,j}$ ，奶牛玩过这个游戏后的产出为  $E_{i,j}$ 。如果想玩同一平台上的多个游戏，只要买一台游戏机就够了。请帮助约翰选择买哪些游戏机和游戏，才能使奶牛的产奶效益之和最大？注意同一个游戏买两次是会产生双倍效益产生的。

### 输入格式

- 第一行：两个整数  $K$  和  $V$ ， $1 \leq K \leq 50, 1 \leq V \leq 10^6$
- 第二行到第  $K+1$  行：第  $i+1$  行首先有两个整数  $C_i$  和  $S_i$ ， $1 \leq C_i \leq 10^6, 1 \leq S_i \leq 10$ ，其次有  $S_i$  对整数  $P_{i,j}$  和  $E_{i,j}$ ， $1 \leq P_{i,j} \leq 10^6, 1 \leq E_{i,j} \leq 10^6$

### 输出格式

- 单个整数：表示可以得到的最大产出之和

### 样例输入

```
3 800
300 2 30 50 25 80
600 1 50 130
400 3 40 70 30 40 35 60
```

### 样例输出

210

### 解释

购买第一种游戏平台上的第二个游戏，以及第三种游戏平台上的第一个和第三个游戏，恰好花去  $300 + 25 + 400 + 40 + 35 = 800$  元，产出为  $80 + 70 + 60 = 210$

来源

Video Game Troubles, 2009 Dec

代号

vidgame

解答



## 112 ★★★★★☆ 三个代表

在最近的威斯康辛州人民代表大会上，奶牛们分到了三个代表名额。威州共有  $3N$  个城市，按照选举流程，应将这些城市划分成三个选区，每个选区通过投票选出一个代表。至于哪些城市归属哪个选区，是由选举委员会决定的，唯一的要求是每个选区应恰好由  $N$  个城市组成。

而然，这里存在一个阴谋。威斯康辛州有两种奶牛——杰尔西牛和荷斯坦牛，彼此明争暗斗，貌合神离。每个城市里，恰好有  $K$  头奶牛参加投票，其中杰尔西牛有  $J_i$  头，剩下的都是荷斯坦牛。杰尔西牛控制了选举委员会，它们正在密谋如何保证自己种群的代表胜出。它们的底线是至少让三个代表中的两个来自杰尔西，这就需要让至少两个选区的杰尔西牛多于荷斯坦牛。请你帮助它们设计一个划分选区的方法，来完成它们的阴谋。保证分法一定存在，如果有多种方案，任意输出一种即可。

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq N \leq 60, 1 \leq K \leq 50000$
- 第二行到第  $3N + 1$  行：第  $i + 1$  行有两个整数  $J_i$ ， $0 \leq J_i \leq K$

### 输出格式

- 第一行：每行  $N$  个整数，表示分在第一个选区内的城市编号
- 第二行：每行  $N$  个整数，表示分在第二个选区内的城市编号
- 第三行：每行  $N$  个整数，表示分在第三个选区内的城市编号

（必须改成 yes/no，输出方案不能用位压缩）

### 样例输入

2  
510  
500  
500  
670  
400  
310

### 样例输出

1 2  
3 6  
5 4

### 解释

如果把第二个城市和第三个城市分成一个选区，这个选区里的杰尔西牛和荷斯坦牛一样多，是不能保证杰尔西牛的一定胜出的

来源

Jersey Politics, 2005 Feb

代号

jpol

解答

## 113 ★★★★★ 最少找零

约翰在镇上买了  $T$  元钱的东西，正在研究如何付钱。假设有  $N$  种钞票，第  $i$  种钞票的面值为  $V_i$ ，约翰身上带着这样的钞票  $C_i$  张。商店老板罗伯是个土豪，所有种类的钞票都有无限张。他们有洁癖，所以希望在交易的时候，交换的钞票张数尽可能地少。请告诉约翰如何恰好付掉  $T$  元，而且在过程中交换的货币数量最少。

### 输入格式

- 第一行：两个整数  $N$  和  $T$ ， $1 \leq N \leq 100, 1 \leq T \leq 10000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $V_i$  和  $C_i$ ， $1 \leq V_i \leq 120, 0 \leq C_i \leq 10000$

（需要修改题目的数据格式）

### 输出格式

- 单个整数：表示付钱找零过程中交换的最少货币数量，如果约翰的钱不够付账，或老板没法找开零钱，输出  $-1$

### 样例输入

3 70

5 25 50

5 2 1

（原来的格式不好，改正为）

3 70

5 5

25 2

50 1

### 样例输出

3

### 解释

约翰付给老板 75 元，老板找约翰 5 元，交换了 3 张钞票

来源

The Fewest Coins, 2006 Dec

代号

fewcoins

解答

## Chapter 36

# 动态规划（六）——树型 DP

## 114 ★★☆☆☆ 圣诞装饰

约翰正在装饰他家的圣诞树。圣诞树上有个  $N$  个结点，第一个结点是根，其余结点都有唯一的父亲结点，第  $i$  个结点的父亲是  $P_i$ 。由于根没有父亲，所以记  $P_1 = -1$ 。

约翰可以在每个结点上挂载装饰物，但费用是变化。在第  $i$  个结点上挂载一个装饰物需要花费  $C_i$  元钱。奶牛对这个圣诞树上每个结点都有特殊的装饰需求，对于第  $i$  个结点，奶牛要求以它为根的子树上必须有  $D_i$  个装饰物。请问约翰在哪些结点上挂载装饰物，才能满足奶牛的要求，并且使得装饰费用最少？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10^5$
- 第二行到  $N$  行：第  $i+1$  行有三个整数： $P_i$ ， $D_i$  和  $C_i$ ， $1 \leq P_i \leq N$ ， $0 \leq D_i \leq 10^6$ ， $1 \leq C_i \leq 100$

### 输出格式

- 单个整数，表示完成所有装饰要求的最少费用

### 样例输入

```
5
-1 9 3
1 2 2
5 3 2
5 1 4
2 3 3
```

### 样例输出

```
20
```

### 解释

在第四个结点上放一个，花 4 元；在第三个结点上放五个，花 10 元；在第二个结点上放三个，花 6 元；

来源

Tree Decoration, 2011 Mar

代号

tdec

解答

## 115 ★★★☆☆ 拜访奶牛

工作多日之后，贝西终于获得了假期。作为牛群里最活跃的奶牛，她决定拜访一下她的朋友们。

贝西有  $N$  位好友，每位好友住的地方都不一样，有  $N - 1$  条道路连接了她们的住所，第  $i$  条道路连接了第  $A_i$  个好友和第  $B_i$  个好友的家， $A_i$  和  $B_i$  被称作是邻居。保证所有好友之间都有道路连通。

约翰想让贝西早点回家，但贝西想多去朋友家玩一会儿，于是他们达成了妥协。贝西可以自由选择去哪些朋友家做客，但如果访问过朋友  $C$  的家，就不能再访问  $C$  的邻居了，因为约翰觉得这些邻居可以到  $C$  的家里和贝西见面。

请问机智的贝西应该选择访问哪些朋友的家，才能让她去过的地方到达最大？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到  $N$  行：第  $i + 1$  行有两个整数： $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数，表示贝西最多能访问几家奶牛

### 样例输入

```
7
6 2
3 4
2 3
1 2
7 6
5 6
```

### 样例输出

```
4
```

### 解释

访问  $\{1, 3, 5, 7\}$ ，也存在其他最优方案



来源

Visiting Cows, 2010 Nov

代号

vacation

解答

## 116 ★★☆☆☆ 手机网络

约翰决定为他的奶牛配备手机，以鼓励她们之间互相交流。奶牛居住在  $N$  片牧场上，约翰需要在这些牧场中选择一些，建立通讯基站。一些牧场之间距离较近，构成邻居关系，一个牧场如果建立了基站，那么它和它的邻居就能收到信号，构成邻居关系的牧场有  $N - 1$  对，任意两个牧场之间都可以通过一些邻居连通。请帮助约翰计算一下，要让所有牧场都收到信号，最少要建几座基站？

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 10000$
- 第二行到  $N$  行：第  $i + 1$  行有两个整数： $A_i$  和  $B_i$ ，表示第  $A_i$  个牧场和第  $B_i$  个牧场是邻居， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数，表示最少的基站数目

### 样例输入

```
5
1 3
5 2
4 3
3 5
```

### 样例输出

```
2
```

### 解释

建在第二个及第三个牧场上，或者建在第三个和第五个牧场也可以

来源

Cell Phone Network, 2008 Jan

代号

tower

解答

## 117 ★★★★★☆ 切断道路

约翰有  $N$  个牛棚，这些牛棚间有  $N - 1$  条双向道路。第  $i$  条道路连接了第  $A_i$  个牧场和第  $B_i$  个牧场， $A_i$  和  $B_i$  被称为邻居。任意两牛棚都可以通过道路连通。

约翰得到消息，洪水将要到来了，于是他打算把奶牛集中安置到避难点。避难点必须由相邻的  $P$  个牛棚组成。他还要切断这些牛棚和其他牛棚之间的道路。请问约翰该选择哪些牛棚作为避难点，才能使得需要切断的道路最少呢？

### 输入格式

- 第一行：两个整数  $N$  和  $P$ ， $1 \leq P \leq N \leq 150$
- 第二行到  $N$  行：第  $i + 1$  行有两个整数： $A_i$  和  $B_i$ ， $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数，表示需要切断的道路数量

### 样例输入

```
11 6
1 2
1 3
1 4
1 5
2 6
2 7
2 8
4 9
4 10
4 11
```

### 样例输出

```
2
```

### 解释

选择  $\{1, 2, 3, 6, 7, 8\}$ ，需要加固的是  $(1, 4)$  和  $(1, 5)$

来源

Rebuilding Roads, Feb 2002

代号

reroads

解答

## 118 ★★★★★ 产奶比赛

约翰的  $N$  头奶牛报名去参加世界产奶大奖赛。这是一项组队赛，队伍的人数任意，约翰的第  $i$  头奶牛可以为集体贡献  $C_i$  升牛奶。注意有些奶牛会帮倒忙——弄翻其他奶牛的牛奶桶，所以  $C_i$  可能是负的。所有参加比赛的奶牛毫无疑问都是女士，记第  $i$  头奶牛的母亲为  $P_i$ ，如果她的母亲没有报名， $P_i$  就等于零。

约翰很清楚对手的实力，确信只要自己队伍产出的牛奶不少于  $X$  升，就一定能够赢得比赛。比赛举办的目的之一，是通过竞赛中的合作来增进奶牛家庭成员之间的默契。奶牛们觉得既然自己稳操胜券，为了表示对大赛精神的支持，她们希望派出的队伍里多出现一些母女组合。请挑选出一只能赢得比赛的队伍，使得其中奶牛的母女关系数目最多。如果一头奶牛和她的母亲、外祖母组成一只队伍，这只队伍算作有两对母女关系。

### 输入格式

- 第一行：两个整数  $N$  和  $X$ ， $1 \leq N \leq 500, 1 \leq X \leq 10^7$
- 第二行到  $N$  行：第  $i+1$  行有两个整数： $C_i$  和  $P_i$ ， $-10000 \leq C_i \leq 10000, 0 \leq P_i \leq N$ ，保证输入中的母女关系合理，不会出现循环

### 输出格式

- 单个整数，表示最大的母女关系数，如果无法选出一支优胜队伍，输出  $-1$

### 样例输入

```
5 8
-1 0
3 1
5 1
-3 3
2 0
```

### 样例输出

```
2
```

### 解释

选  $\{1, 2, 3, 5\}$ ，产出为  $-1+3+5+2=9 \geq 8$ 。这支队伍里有两对母女关系  $(1, 2)$  和  $(1, 3)$ 。如果选  $\{2, 3, 5\}$ ，虽然产奶量更高，但这个队伍的母女关系数为 0。

## 来源

Milk Team Select, 2006 Mar

## 代号

tselect

## 解答

## 119 ★★★★★ 焊接

贝西可以用电线焊接出很多图形。焊接所用的电线要从商店里买，各种长度的电线无限量，长度为  $L$  的电线售价为  $L^2$ 。为了安全，任何情况下都不能把长电线剪短，也不能把两根电线的两个端点焊接起来冒充一条长电线。

两条电线焊接在一起的方法是：加热一条电线的端点，将它焊死在另一条电线的中间位置上。注意不能把两条电线的端点焊死在一起，这是变相把两根短电线接长了，违反了操作安全。加热只能在电线的端点处进行，所以只用两条电线的话是不可能焊出 X 字形图案的。如果想焊接出一个 X 字形，可以用三根电线。前两根先焊出一个 Y 字形，再把最后一根电线的端点烧热，焊接在 Y 字形的三叉交点上。

约翰提供了一个连通图形，有  $N$  个点， $N-1$  条边，每条边的长度都是 1，希望贝西能按照这个图形焊接电线。贝西乐意效劳，请问她如何选择焊接的材料和顺序，才能使购买电线的总费用最小。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到  $N$  行：第  $i+1$  行有两个整数： $A_i$  和  $B_i$ ，表示图形中的一条边  $1 \leq A_i, B_i \leq N$

### 输出格式

- 单个整数，表示最少花费，保证这个数字小于  $2^{63}$

### 样例输入

```
6
1 2
1 3
1 4
1 5
1 6
```

### 样例输出

```
7
```

### 解释

每个点都和 1 相连，因此只要购买一条长度为 2 的电线和三条长度为 1 的电线即可。总花费为  $2^2 + 3 \times 1^2$



来源

Soldering, 2011 Open

代号

solder

解答

## Chapter 37

# 动态规划（七）——单调队列优化

## 120 ★★★☆☆ 修剪草坪

约翰让他的奶牛来修建草坪。他有  $N$  头奶牛，第  $i$  头奶牛的工作能力为  $A_i$ 。编号相近的奶牛很熟悉，如果同时让  $K + 1$  头编号连在一起的奶牛工作，她们就会密谋罢工。请问，约翰应该让哪些奶牛同时工作，使得它们的能力之和最大，而且不会罢工。

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $1 \leq K \leq N \leq 10^5$
- 第二行到  $N + 1$  行：第  $i + 1$  行有一个整数  $A_i$ ， $1 \leq A_i \leq 10^9$

### 输出格式

- 单个整数，表示在所有不会罢工的奶牛组合之中，最大的能力之和

### 样例输入

```
5 2
1
2
3
4
5
```

### 样例输出

```
12
```

### 解释

除了第三头以外的所有奶牛都工作，总能力为  $1 + 2 + 4 + 5 = 12$

## 来源

Mowing the Lawn, 2011 Open

2010 年提高组初赛程序填空。烽火传递。几乎一样

## 代号

mowlawn

## 解答

## 121 ★★★★★☆ 玉米实验

约翰决定培育新的玉米品种以提高奶牛的产奶效率。约翰所有的土地被分成  $N \times N$  块，其中第  $r$  行第  $c$  列的玉米质量为  $A_{r,c}$ 。他打算找一块质量比较均匀的土地开始自己的实验。经过前期考察，他已经锁定了  $K$  片区域作为实验基地的候选，其中第  $i$  片区域是从  $R_i$  行  $C_i$  列开始，到  $R_i + B - 1$  行  $C_i + B - 1$  列结束的一个  $B \times B$  的区域。请帮助约翰计算一下，在这些候选区域里，玉米的最高质量与最低质量之差分别是多少。

### 输入格式

- 第一行：三个整数  $N$ ， $B$  和  $K$ ， $1 \leq B \leq N \leq 250$ ， $1 \leq K \leq 10^5$
- 第二行到  $N + 1$  行：第  $i + 1$  行有  $N$  个整数，分别表示  $A_{i,1}$  到  $A_{i,N}$ ，对所有  $1 \leq j \leq N$ ，满足  $0 \leq A_{i,j} \leq 250$
- 第  $N + 2$  行到第  $N + K + 1$  行：第  $N + i + 1$  行有两个整数  $R_i$  和  $C_i$ ， $1 \leq R_i, C_i \leq N$

### 输出格式

- 第一行到第  $K$  行：第  $i$  行表示第  $i$  片候选区域中最高质量与最低质量之差

### 样例输入

```
5 3 1
5 1 2 6 3
1 3 5 2 7
7 2 4 6 1
9 9 8 6 5
0 6 9 3 9
1 2
```

### 样例输出

```
5
```

### 解释

最高为 6，最低为 1，所以差是 5

## 来源

Cornfields, 2003 Mar

## 代号

cornfld

## 122 ★★★★★☆ 干草金字塔

贝西要用干草包堆出一座金字塔。干草包会从传送带上陆续运来，依次出现  $N$  包，每包干草可以看做是一个二维平面上的一个长方形，第  $i$  包干草的宽度是  $W_i$ ，长度统一为 1。

金字塔的修建有几个规定，首先，为了建筑稳定，塔一定要形成类似“金”字的样子，即塔的上层宽度不能超过下层宽度，而且每层的干草包必须紧靠在一起，不能出现缝隙。其次，由于干草是陆续送来的，所以先送来的干草放在较低层。贝西会选择最先送来的几包干草，堆在地上作为第一层，然后再把紧接着送来的几包干草包放在第二层，再铺建第三层……重复这个过程，一直到所有的干草全部用完。最后，贝西不喜欢浪费，所有干草包一定要用上，不能弃置不用。贝西的目标是建一座最高的金字塔，在遵循上述规定的前提下，她可以任意决定在金字塔的每一层布置多少连续的干草包。请你来帮助她完成这个任务吧。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 100000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $W_i$ ， $1 \leq W_i \leq 10000$

### 输出格式

- 单个整数：表示可以建成的最高高度

### 样例输入

3  
1  
2  
3

### 样例输出

2

### 解释

将 1 和 2 放在第一层，将 3 放在第二层

## 来源

Tower of Hay, 2009 Open

## 代号

pyramid

原代号是 tower，与另一题重复，注意修改

## 分析

### $O(N^3)$ 的做法

设  $h[i, j]$  为第  $i$  块到第  $j$  块作为底座的最高高度

$$h[i, j] = \max_{w_i + w_{i+1} + \dots + w_j \geq w_{j+1} + w_{j+2} + \dots + w_k} \{h[j+1, k]\} + 1$$

### $O(N^2)$ 的做法

主要讲如何证明最瘦的塔一定是最高的塔。鸽笼原理。设最瘦的分四段，较胖的分五段。五段需要四个分隔符，不能放在最瘦的第一段，必须在后面三段里放四个分割符，必有一段放两个分割符，此时，后面的分发粘给最瘦的，可以形成一个更好的解法。

设  $f[i]$  为从第  $i$  块到第  $N$  块组成塔时的最小宽度

$$f[i] = \min_{S_{j-1} - S_{i-1} \geq f[j]} \{S_{j-1} - S_{i-1}\}$$

### $O(N)$ 的做法

单调栈



## 123 ★★★★★☆ 又买饲料

约翰开车回家，又准备顺路买点饲料了（咦？为啥要说“又”字？）回家的路程一共有  $E$  公里，这一路上会经过  $K$  家商店，第  $i$  家店里有  $F_i$  吨饲料，售价为每吨  $C_i$  元。约翰打算买  $N$  吨饲料，他知道商家的库存是足够的，至少所有店的库存总和不会少于  $N$ 。除了购买饲料要钱，运送饲料也是要花油钱的，约翰的卡车上如果装着  $X$  吨饲料，那么他行驶一公里会花掉  $X^2$  元，行驶  $D$  公里需要  $DX^2$  元。已知第  $i$  家店距约翰所在的起点有  $X_i$  公里，那么约翰在哪些商店买饲料运回家，才能做到最省钱呢？

### 输入格式

- 第一行：三个整数  $K$ ， $E$  和  $N$ ， $1 \leq K \leq 10000, 1 \leq E \leq 500, 1 \leq N \leq 500$
- 第二行到第  $N+1$  行：第  $i+1$  行有三个整数  $X_i$ ， $F_i$  和  $C_i$ ， $0 < X_i < E, 1 \leq F_i \leq 10000, 1 \leq C_i \leq 10^7$

### 输出格式

- 单个整数：表示购买及运送饲料的最小费用

### 样例输入

```
2 5 3
3 1 2
4 1 2
1 1 1
```

### 样例输出

9

### 解释

在离家较近的两家商店里各购买一吨饲料，则花在路上的钱是  $1 + 4 = 5$ ，花在店里的钱是  $2 + 2 = 4$

来源

Buying Feed, 2010 Nov

代号

feedsqr

解答

## 124 ★★★★★ 土地并购

约翰准备扩大他的农场，眼下必须购买  $N$  块长方形的土地。如果约翰单买一块土地，价格就是土地的面积。他也可以选择并购一组土地，并购的价格为这些土地中最大的长乘以最大的宽。比如约翰并购一块  $3 \times 5$  和一块  $5 \times 3$  的土地，他只需要支付  $5 \times 5 = 25$  元，比分开买合算。约翰发现，将这些土地分成不同的小组来并购可以节省经费。给定每份土地的尺寸，请你帮助他计算购买所有土地所需的最小费用。

### 输入格式

- 第一行：单个整数  $N$ ， $1 \leq N \leq 50000$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有两个整数  $H_i$  和  $W_i$ ， $1 \leq H_i, W_i \leq 10^6$

### 输出格式

- 单个整数：购买所有土地的最小费用

### 样例输入

```
4
100 1
15 15
20 5
1 100
```

### 样例输出

```
500
```

### 解释

分三组：第一组  $100 \times 1$ ，第二组  $1 \times 100$ ，第三组是  $20 \times 5$  和  $15 \times 15$ ，价格分别为 100、100 和 300

来源

Land Acquisition, 2008 Mar

代号

acquire

解答

## 125 ★★★★★ 约翰的书架

你们一定知道约翰整天都忙着挤牛奶、堆干草、修栅栏、牵奶牛，但你们一定不知道其实约翰还是一个文艺青年。他空的时候会静下心来读书。他收集了  $N$  本书，每本书都有自己的高度和宽度，第  $i$  本书的宽度是  $W_i$ ，高度是  $H_i$ 。

约翰想做个书架来放置这套书。书架是长方形的，分为若干层，具体分多少层由约翰自由决定。由于木料的问题，书架的宽度固定为  $L$ ，即在一层书架上的书的宽度之和不能超过  $L$ 。

既然书架的宽度是固定的，约翰想让书架的高度尽量低一点。这些书构成一个系列，为了检索方便，书必须按照编号顺序摆放。也就是说，第一本书必须放在第一层，同层的书籍次序必须连续，如果一本书是第  $k$  层书架上最后一本，那么下一本就必须放在第  $k+1$  层上。

假设书架的高度就是每层书架的高度之和，每层书架的高度是该层中最高的一本书。请问约翰应该选择一个多少层次的书架，每层书架放哪些书，才能让书架的高度最小？

### 输入格式

- 第一行：两个整数  $N$  和  $L$ ， $1 \leq N \leq 100000, 1 \leq L \leq 10^9$
- 第二行到第  $N+1$  行：第  $i+1$  行有两个整数  $H_i$  和  $W_i$ ， $1 \leq H_i \leq 10^6, 1 \leq W_i \leq L$

### 输出格式

- 单个整数：表示书架的最低高度

### 样例输入

```
5 10
5 7
9 2
8 5
13 2
3 8
```

### 样例输出

```
21
```

### 解释

将书架做成三层，第一层放第一本，第二层放第二本到第四本，第三层放第五本

## 来源

Bookshelf, 2012 Open

## 代号

bookshelf

## Chapter 38

# 动态规划（八）——位向量型

## 126 ★★★☆☆ 混乱奶牛

约翰有  $N$  头奶牛，第  $i$  头奶牛的编号是  $S_i$ ，每头奶牛的编号都不同。这些奶牛最近在闹脾气，为表达不满的情绪，她们在排队的时候一定要排成混乱的队伍。如果一只队伍里所有位置相邻的奶牛的编号之差都大于等于  $K$ ，那么这就是一只混乱的队伍，其中  $K$  是一个给定的整数。比如说，当  $K = 2$  时，序列  $(1, 3, 5, 2, 6, 4)$  就是一只混乱的队伍，而  $(1, 3, 6, 5, 2, 4)$  不是，因为 6 和 5 只差 1，不够混乱。请问，这  $N$  头奶牛可以排成多少种混乱的队形呢？

### 输入格式

- 第一行：两个整数  $N$  和  $K$ ， $4 \leq N \leq 16$ ， $1 \leq K \leq 3400$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有一个整数  $S_i$ ， $1 \leq S_i \leq 25000$

### 输出格式

- 单个整数：表示混乱队伍的数量

### 样例输入

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

### 样例输出

```
2
```

### 解释

两种排法是 3, 1, 4, 2 和 2, 4, 1, 3



来源

Mixed Up Cows, USACO 2008 Nov

代号

mixup

解答

## 127 ★★★☆☆ 全能奶牛

约翰有  $N$  头奶牛，组成了一支队伍参加全能比赛。比赛一共有  $N$  项，每头奶牛必须参加一项比赛，每项比赛也必须有一头奶牛参加。任何一头奶牛可以胜任任何一项比赛，但得分不一样。如果第  $i$  头奶牛参加第  $j$  项比赛，在比赛结束的时候，可以为团体总分增加  $S_{i,j}$ 。

比赛是按照顺序依次进行的。除了上述获得分数的方法之外，还有  $B$  种奖励分。获得奖励的方法是在前几项比赛里获得足够的分数。具体来说，第  $i$  项奖励会在第  $K_i$  项比赛结束的时候检查，如果当时的总分大于或等于  $P_i$ ，奶牛们就可以立即获得额外的  $A_i$  分。如果有多项奖励在同一时刻检查，奶牛可以自由安排检查和加分的顺序。请问约翰应该如何安排奶牛参加比赛，才能让它们获得最高的分数？

### 输入格式

- 第一行：两个整数  $N$  和  $B$ ， $1 \leq N \leq 20, 1 \leq B \leq 20$
- 第二行到第  $B + 1$  行：第  $i + 1$  行有三个整数  $K_i$ ， $P_i$  和  $A_i$ ， $1 \leq K_i \leq N, 1 \leq P_i \leq 40000, 1 \leq A_i \leq 1000$
- 第  $B + 2$  行到第  $B + N + 1$  行：第  $i + B + 1$  行有  $N$  个整数，代表  $S_{i,1}$  到  $S_{i,N}$ ，对每个  $1 \leq j \leq N, 1 \leq S_{i,j} \leq 1000$

### 输出格式

- 单个整数：表示奶牛们可以获得的最大得分

### 样例输入

```
3 1
2 7 6
5 1 7
2 2 4
4 2 1
```

### 样例输出

```
17
```

### 解释

第一项比赛由第一头奶牛参加，第二项比赛由第三头奶牛参加，第三项比赛由第二头奶牛参加

## 来源

Cow Decathlon, 2014 Feb

## 代号

dec

## 解答

## 参考程序

```
1  // todo
1  const
2    maxn = 20;
3    maxb = 20;
4  var
5    sk : array[1..maxn, 1..maxn] of longint;
6    k, p, a : array[1..maxb] of longint;
7    f : array[0..1 shl maxn] of longint;
8    n, b, i, j, s, t, c, mask : longint;
9
10 function popcount(n:longint) : longint;
11 begin
12   popcount := 0;
13   while n > 0 do begin
14     inc(popcount, n and 1);
15     n := n shr 1;
16   end;
17 end;
18
19 begin
20   read(n, b);
21   for i := 1 to b do
22     read(k[i], p[i], a[i]);
23   for i := 1 to b do
24     for j := i + 1 to b do
25       if p[i] > p[j] then begin
26         t := k[i]; k[i] := k[j]; k[j] := t;
27         t := p[i]; p[i] := p[j]; p[j] := t;
28         t := a[i]; a[i] := a[j]; a[j] := t;
29       end;
30   for i := 1 to n do
31     for j := 1 to n do
32       read(sk[i,j]);
33   for s := 1 to (1 shl n) - 1 do begin
34     c := popcount(s);
35     for i := 1 to n do begin
36       mask := 1 shl (i - 1);
37       if s and mask > 0 then begin
38         t := f[s - mask] + sk[i, c];
39         if f[s] < t then f[s] := t;
40       end;
41     end;
42     for j := 1 to b do
43       if (k[j] = c) and (f[s] >= p[j]) then
44         inc(f[s], a[j]);
45   end;
46   writeln(f[(1 shl n) - 1]);
47 end.
```

## 128 ★★★★★☆ 洞穴奶牛

贝西喜欢去洞穴探险。这次她去的地方由  $N$  个洞穴组成，编号分别是 1 到  $N$ ，1 号洞穴是出发的起点。

洞穴之间由  $M$  条隧道相连，双向通行，第  $i$  条隧道连接  $A_i$  和  $B_i$ 。每条隧道有个尺寸，第  $i$  条隧道的尺寸为  $W_i$ ，如果贝西的身材超过了隧道的尺寸，就钻不过去了。贝西的身材可用一个数字来描述，一开始是 0。

前期准备时，贝西已在  $K$  个里洞穴存放了食物，她当然不会放过它们。当她吃掉一份食物后，身材就会增加 1。在探险的过程中，她想吃掉尽量多的食物，但要保证最后能返回起点。请帮她计划一下，什么样的路线才能吃到最多的食物？

注意，探险的路线可以重复，所以路过有食物时，可以留到下次路过的时候再吃。

### 输入格式

- 第一行：三个整数  $N$ ， $M$  和  $K$ ， $1 \leq N \leq 100, 1 \leq M \leq 1000, 1 \leq K \leq 14$
- 第二行到第  $K+1$  行：第  $i+1$  行有一个整数  $C_i$ ，表示在第  $C_i$  号洞穴里有一份食物， $1 \leq C_i \leq N$
- 第  $K+2$  行到第  $K+M+1$  行：第  $i+K+1$  行有三个整数  $A_i$ ， $B_i$  和  $W_i$ ， $1 \leq A_i, B_i \leq N, 1 \leq W_i \leq 100$

### 输出格式

- 单个整数：表示贝西最多能吃几份食物

### 样例输入

```
6 7 5
1
2
3
4
5
1 2 3
3 6 2
6 2 10
2 4 1
5 1 1
4 5 1
1 6 1
```

### 样例输出

```
4
```

### 解释

先吃 5 号洞里的，再吃 3，2，最后走向 1，吃掉 1 里的最后一份食物

来源

Cave Cow 1, 2004 Open

代号

cavecow

解答

## 129 ★★☆☆☆ 不找零

约翰带着  $N$  头奶牛在超市买东西，现在他们正在排队付钱，排在第  $i$  个位置的奶牛需要支付  $C_i$  元。今天说好所有东西都是约翰请客的，但直到付账的时候，约翰才意识到自己没带钱，身上只有  $K$  张消费卡，第  $i$  张卡里有  $V_i$  元余额。

问题是，这些消费卡都是一次性的，它们可以被收银机读取，但如果卡一旦离开了收银机，卡里的余额就会归零，而且超市也不负责找零！奶牛的队伍很长，不可能再调整她们的位置了，所以一张卡只能支付一段连在一起的账单。而且，一张账单只能用一张消费卡支付，超市的系统不接受用两张或以上的卡支付一笔账单。

约翰的问题就是按照什么样的顺序来使用这些消费卡，才能让他能为所有的奶牛买单，而且使得剩余的消费卡的余额之和最大呢？

### 输入格式

- 第一行：两个整数  $K$  和  $N$ ， $1 \leq K \leq 16$ ,  $1 \leq N \leq 10^5$
- 第二行到第  $K+1$  行：第  $i+1$  行有一个整数  $V_i$ ， $1 \leq V_i \leq 10^9$
- 第  $K+2$  行到第  $K+N+1$  行：第  $i+K+1$  行有一个整数  $C_i$ ， $1 \leq C_i \leq 10^4$

### 输出格式

- 单个整数：表示约翰买完所有奶牛的单之后，最多还能剩多少余额，如果他带的卡根本没有办法支付所有的账单，输出  $-1$ 。

### 样例输入

```
3 6
12
15
10
6
3
3
2
3
7
```

### 样例输出

```
12
```

### 解释

用 10 元的卡支付前两笔账单，然后用 15 元的卡支付后面所有的账单，还剩下一张 12 元的卡没用

来源

No Change, 2013 Nov

代号

nochange

解答

## 130 ★★★★★☆ 奶牛乘电梯

有  $N$  头奶牛需要乘电梯上楼，第  $i$  头奶牛重  $W_i$ ，电梯的载重上限是  $C$ ，请问电梯最少往返几次才能运完所有的奶牛？

### 输入格式

- 第一行：两个整数  $N$  和  $C$ ， $1 \leq N \leq 18, 1 \leq C \leq 10^8$
- 第二行到第  $N+1$  行：第  $i+1$  行有一个整数  $W_i$ ， $1 \leq W_i \leq C$

### 输出格式

- 单个整数：表示电梯的最少往返次数

### 样例输入

4 10  
5  
6  
3  
7

### 样例输出

3

### 解释

5 和 3 一组，6 单独一组，7 单独一组



来源

Cows in a Skyscraper, 2012 Mar

代号

skyscraper

解答

## 131 ★★★★★☆ 安排牧场

约翰有一片土地，这片土地可以分成  $N \times M$  个方格，除了一些贫瘠的方格之外，每个方格可以放牧一头奶牛。奶牛们要求有一种独霸的感觉，两头奶牛的地盘如果共享一条边界，那么它们就会打架。约翰还没决定在哪些方格上放牧奶牛。作为参考，他想知道一共有多少种方案可供选择，请你帮他算一下，有多少种可以避免奶牛打架的方案。由于方案可能很多，请输出方案数模  $10^8$  的余数。

### 输入格式

- 第一行：两个整数  $N$  和  $M$ ， $1 \leq N, M \leq 12$
- 第二行到第  $N + 1$  行：第  $i + 1$  行有  $M$  个整数  $C_{i,1}$  到  $C_{i,M}$ ，表示第  $i$  行方格中土地的性质，如果  $C_{i,j} = 0$ ，说明这是块贫瘠的土地，不能放牧，如果  $C_{i,j} = 1$ ，说明这块土地可以放牧

### 输出格式

- 单个整数：表示方案数模  $10^8$  的余数

### 样例输入

```
2 3
1 1 1
0 1 0
```

### 样例输出

9

### 解释

如果只放一头奶牛，有 4 种方案；如果放两头奶牛，有 3 种方案；如果放三头奶牛，有 1 种方案；一头都不放，是最后一种方案

## 来源

Corn Fields, 2006 Nov

## 代号

cowfood

## $O(2^{NM})$ 的算法

### 参考程序

```
1  const
2    maxn = 12;
3  var
4    n, m, ans, i, j : longint;
5    field : array[0..maxn, 0..maxn] of longint;
6    { 00  %}
7  procedure search(x, y : longint);
8  begin
9    if x > n then inc(ans)
10   else begin
11     if y < m then
12       search(x, y + 1)
13     else
14       search(x + 1, 1);
15     if (field[x, y] = 1)
16       and (field[x-1, y] >= 0)
17       and (field[x, y-1] >= 0) then begin
18       field[x, y] := -1;
19       if y < m then
20         search(x, y + 1)
21       else
22         search(x + 1, 1);
23       field[x, y] := 1; { 2»i}
24     end;
25   end;
26 end;
27
28 begin
29   read(n, m);
30   for i := 1 to n do
31     for j := 1 to m do
32       read(field[i, j]);
33   search(1, 1);
34   writeln(ans);
35 end.
```

## $O(N4^M)$ 的算法

逐行递推

## $O(NM2^M)$ 的算法

逐格递推