IOI2013国家集训队作业
CodeForces解题报告

湖南师大附中彭天翼
CODEFORCES用户ID: pty
完成情况: 108/108

January 13, 2013

Contents

1 codeforces(1) 10

1.1 7E Defining Macros . . . 10
1.1.1 Description . . . 10
1.1.2 Analysis . . . 10
1.2 8D Two Friends . . . 12
1.2.1 Description . . . 12
1.2.2 Analysis . . . 12
1.3 8E Beads . . . 14
1.3.1 Description . . . 14
1.3.2 Analysis . . . 14
1.4 10E Greedy Change . . . 15
1.4.1 Description . . . 15
1.4.2 Analysis . . . 15
1.5 15E Triangles . . . 18
1.5.1 Description . . . 18
1.5.2 Analysis . . . 19
1.6 17C Balance . . . 20
1.6.1 Description . . . 20
1.6.2 Analysis . . . 20
1.7 17E Palisection . . . 21
1.7.1 Description . . . 21
1.7.2 Analysis . . . 21
1.8 19E fairy . . . 22
1.8.1 Description . . . 22
1.8.2 Analysis . . . 22
1.9 23D Tetragon . . . 23
1.9.1 Description . . . 23
1.9.2 Analysis . . . 23
1.10 23E Tree . . . 24

1.10.1 Description 24
1.10.2 Analysis 24

2 codeforces(2) 25
2.1 26E Multithreading 25
2.1.1 Description 25
2.1.2 Analysis 25
2.2 28D Don't fear 27
2.2.1 Description 27
2.2.2 Analysis 27
2.3 30D King's Problem 28
2.3.1 Description 28
2.3.2 Analysis 28
2.4 30E Password 29
2.4.1 Description 29
2.4.2 Analysis 29
2.5 32E Hide-and-Seek 30
2.5.1 Description 30
2.5.2 Analysis 30
2.6 35E Parade 31
2.6.1 Description 31
2.6.2 Analysis 31
2.7 36E Two Path 32
2.7.1 Description 32
2.7.2 Analysis 32
2.8 37E Trial for Chief 33
2.8.1 Description 33
2.8.2 Analysis 33
2.9 39C Moon Craters 34
2.9.1 Description 34
2.9.2 Analysis 34
2.10 39A Calculations 35
2.10.1 Description 35
2.10.2 Analysis 35

3 codeforces(3) 36
3.1 39E Dirichlet 36
3.1.1 Description 36
3.1.2 Analysis 36
3.2 40E Number Table 37

3.2.1 Description . . . 37
3.2.2 Analysis . . . 37
3.3 43E Race . . . 38
3.3.1 Description . . . 38
3.3.2 Analysis . . . 38
3.4 44J Triminoes . . . 39
3.4.1 Description . . . 39
3.4.2 Analysis . . . 39
3.5 45G Prime . . . 40
3.5.1 Description . . . 40
3.5.2 Analysis . . . 40
3.6 45E Director . . . 41
3.6.1 Description . . . 41
3.6.2 Analysis . . . 41
3.7 46F Hercule Poirot . . . 42
3.7.1 Description . . . 42
3.7.2 Analysis . . . 42
3.8 47E cannon . . . 43
3.8.1 Description . . . 43
3.8.2 Analysis . . . 43
3.9 49E Ancestor . . . 44
3.9.1 Description . . . 44
3.9.2 Analysis . . . 44
3.10 51F Caterpillar . . . 45
3.10.1 Description . . . 45
3.10.2 Analysis . . . 45

4 codeforces(4) 46
4.1 53E E.Dead Ends . . . 46
4.1.1 Description . . . 46
4.1.2 Analysis . . . 46
4.2 57D D.Journey . . . 46
4.2.1 Description . . . 46
4.2.2 Analysis . . . 47
4.3 217D Bitonix . . . 48
4.3.1 Description . . . 48
4.3.2 Analysis . . . 48
4.4 67E Save the City . . . 49
4.4.1 Description . . . 49
4.4.2 Analysis . . . 49

4.5 67C Sequence of Balls 50
4.5.1 Description 50
4.5.2 Analysis 50
4.6 70D Professor's task 51
4.6.1 Description 51
4.6.2 Analysis 51
4.7 70E Reform 52
4.7.1 Description 52
4.7.2 Analysis 52
4.8 156E Pancakes 53
4.8.1 Description 53
4.8.2 Analysis 53
4.9 105D Geodetics 54
4.9.1 Description 54
4.9.2 Analysis 54
4.10 226E Noble Knight's Path 55
4.10.1 Description 55
4.10.2 Analysis 55

5 codeforces(5) 56
5.1 193D Two Segments 56
5.1.1 Description 56
5.1.2 Analysis 56
5.2 76A Gift 58
5.2.1 Description 58
5.2.2 Analysis 58
5.3 76F Tourist 59
5.3.1 Description 59
5.3.2 Analysis 59
5.4 77E Martian Food 60
5.4.1 Description 60
5.4.2 Analysis 60
5.5 79D Password 63
5.5.1 Description 63
5.5.2 Analysis 63
5.6 81E Pairs 64
5.6.1 Description 64
5.6.2 Analysis 64
5.7 82E Corridor 65
5.7.1 Description 65

5.7.2 Analysis . . . 65
5.8 83E Subsequences . . . 66
5.8.1 Description . . . 66
5.8.2 Analysis . . . 66
5.9 85E Guard Towers . . . 67
5.9.1 Description . . . 67
5.9.2 Analysis . . . 67

6 codeforces(6) 68
6.1 86E Long Sequence . . . 68
6.1.1 Description . . . 68
6.1.2 Analysis . . . 68
6.2 89D Space Mines . . . 69
6.2.1 Description . . . 69
6.2.2 Analysis . . . 69
6.3 91D Grocer's Problem . . . 69
6.3.1 Description . . . 69
6.3.2 Analysis . . . 70
6.4 93D Flag . . . 71
6.4.1 Description . . . 71
6.4.2 Analysis . . . 71
6.5 97C Winning Strategy . . . 72
6.5.1 Description . . . 72
6.5.2 Analysis . . . 72
6.6 97A Domino . . . 73
6.6.1 Description . . . 73
6.6.2 Analysis . . . 73
6.7 98C Help Greg . . . 74
6.7.1 Description . . . 74
6.7.2 Analysis . . . 74
6.8 98D Help Monks . . . 76
6.8.1 Description . . . 76
6.8.2 Analysis . . . 76
6.9 191D Metro Scheme . . . 77
6.9.1 Description . . . 77
6.9.2 Analysis . . . 77
6.10 164D Minimum Diameter . . . 78
6.10.1 Description . . . 78
6.10.2 Analysis . . . 78

7 codeforces(7) 79
7.1 150E Freezing . . . 79
7.1.1 Description . . . 79
7.1.2 Analysis . . . 79
7.2 101E Candies . . . 80
7.2.1 Description . . . 80
7.2.2 Analysis . . . 80
7.3 103E Buying Sets . . . 81
7.3.1 Description . . . 81
7.3.2 Analysis . . . 81
7.4 105E Lift and Throw . . . 82
7.4.1 Description . . . 82
7.4.2 Analysis . . . 82
7.5 107D Crime . . . 83
7.5.1 Description . . . 83
7.5.2 Analysis . . . 83
7.6 115D Expression . . . 84
7.6.1 Description . . . 84
7.6.2 Analysis . . . 84
7.7 120I Numbers . . . 86
7.7.1 Description . . . 86
7.7.2 Analysis . . . 86
7.8 123E Maze . . . 87
7.8.1 Description . . . 87
7.8.2 Analysis . . . 87
7.9 125E MST Company . . . 89
7.9.1 Description . . . 89
7.9.2 Analysis . . . 89

8 codeforces(8) 90
8.1 193E Fibonacci Number . . . 90
8.1.1 Description . . . 90
8.1.2 Analysis . . . 90
8.2 145D Lucky Pair . . . 92
8.2.1 Description . . . 92
8.2.2 Analysis . . . 92
8.3 132E England . . . 93
8.3.1 Description . . . 93
8.3.2 Analysis . . . 93
8.4 138D World of Darkraft . . . 94

8.4.1 Description 94
8.4.2 Analysis 94
8.5 140F Snowflake 95
8.5.1 Description 95
8.5.2 Analysis 95
8.6 147B Smile House 96
8.6.1 Description 96
8.6.2 Analysis 96
8.7 217E Alien DNA 96
8.7.1 Description 96
8.7.2 Analysis 97
8.8 152D Frames 97
8.8.1 Description 97
8.8.2 Analysis 97
8.9 135E Subsequence 98
8.9.1 Description 98
8.9.2 Analysis 98
8.10 183D T-shirt 99
8.10.1 Description 99
8.10.2 Analysis 99

9 codeforces(9) 100
9.1 163D Large Refrigerator 100
9.1.1 Description 100
9.1.2 Analysis 100
9.2 167E Wizards and Bets 101
9.2.1 Description 101
9.2.2 Analysis 101
9.3 232D Fence 102
9.3.1 Description 102
9.3.2 Analysis 102
9.4 175E Power Defence 103
9.4.1 Description 103
9.4.2 Analysis 103
9.5 176D Hyper string 104
9.5.1 Description 104
9.5.2 Analysis 104
9.6 178F2 Representative 105
9.6.1 Description 105
9.6.2 Analysis 105

9.7 178E3 The Beaver's Problem II 106
9.7.1 Description 106
9.7.2 Analysis 106
9.8 185D Visit of the Great 108
9.8.1 Description 108
9.8.2 Analysis 108
9.9 187D BRT Contract 109
9.9.1 Description 109
9.9.2 Analysis 109
9.10 180B Divisibility Rules 110
9.10.1 Description 110
9.10.2 Analysis 110

10 codeforces(10) 111
10.1 198E Gripping Story 111
10.1.1 Description 111
10.1.2 Analysis 111
10.2 176E Archaeology 112
10.2.1 Description 112
10.2.2 Analysis 112
10.3 196D Good String 113
10.3.1 Description 113
10.3.2 Analysis 113
10.4 200E Tractor College 114
10.4.1 Description 114
10.4.2 Analysis 114
10.5 200A Cinema 115
10.5.1 Description 115
10.5.2 Analysis 115
10.6 201D Brand 116
10.6.1 Description 116
10.6.2 Analysis 116
10.7 201E Organization 117
10.7.1 Description 117
10.7.2 Analysis 117
10.8 204E Little Elephant 119
10.8.1 Description 119
10.8.2 Analysis 119
10.9 207B Military Trainings 120
10.9.1 Description 120

10.9.2 Analysis . . . 120
10.10 207A Beaver's Calculator . . . 121
10.10.1 Description . . . 121
10.10.2 Analysis . . . 121

11 codeforces(11) 122
11.1 167D Wizards and Roads . . . 122
11.1.1 Description . . . 122
11.1.2 Analysis . . . 122
11.2 209C Trails and Glades . . . 124
11.2.1 Description . . . 124
11.2.2 Analysis . . . 124
11.3 212B Good Substrings . . . 125
11.3.1 Description . . . 125
11.3.2 Analysis . . . 125
11.4 212D Cutting a Fence . . . 125
11.4.1 Description . . . 125
11.4.2 Analysis . . . 125
11.5 212C Cowboys . . . 126
11.5.1 Description . . . 126
11.5.2 Analysis . . . 126
11.6 213E Two Permutations . . . 127
11.6.1 Description . . . 127
11.6.2 Analysis . . . 127
11.7 217C Formurosa . . . 127
11.7.1 Description . . . 127
11.7.2 Analysis . . . 127
11.8 229E Gifts . . . 128
11.8.1 Description . . . 128
11.8.2 Analysis . . . 128

12 codeforces(12) 129
12.1 113D Museum . . . 129
12.1.1 Description . . . 129
12.1.2 Analysis . . . 129
12.2 75E Ship's Shortest Path . . . 131
12.2.1 Description . . . 131
12.2.2 Analysis . . . 131

Chapter 1

codeforces(1)

1.1 7E Defining Macros

1.1.1 Description

题意：相信大家都知道C语言预处理指令的define，define用来定义宏。宏其实是在编译之前将指定token替换为对应的表达式。这个问题中不考虑带参数的宏。

例如宏sum的定义为x+y，那么程序中的sum会被替换为x+y，但注意asumb不会被替换为ax+yb，因为这里完整的token是asumb，而不是sum。

宏的定义中可以包含+、-、*、/四种运算符（四则运算，-不能表示负号）、非负整数常量、变量、圆括号和之前定义过的宏。

程序员通常利用宏来减少代码量，但使用不当会无法达到预期效果。例如宏sum被定义为x+y，那么表达式2*sum替换后会变为2*x+y，这时候就需要将sum定义为(x+y)，才能使得替换后变为2*(x+y)，以达到预期效果。

现在按照顺序给出N个宏的定义，最后给出一个表达式，问能否达到预期的效果。预期的效果是每个宏都会被当做整体被运算。

数据范围：T≤10, N≤100, 每一行长度不超过100

难度：★★

1.1.2 Analysis

算法：表达式处理

分析：我们定义出一个表达式的优先级为它最后运算的那个符号，如果是$+-$，优先级就是0，如果是$*/$，优先级就是1，如果最外层有至少一对括号，优先级就是2，特别的，如果这个表达式本身就不能达到预期，那么优先级是-1。现在我们可以按如下方式递归的解决这个问题：

1. 找到这个表达式最后运算的符号
2. 如果最外层有括号，该表达式优先级为2
3. 如果这个符号是$+-$，则我们递归两侧的表达式，如果两侧中有-1，或者当右侧优先级为0，且本层符号为$-$时，该表达式优先级为-1, 否则为0
4. 如果这个符号是$*/$，同样递归两侧的表达式，如果两侧中有-1，或者有0，或者当右侧优先级为1，且本层符号为/时，该表达式优先级为-1，否则为1

对于#define的表达式，我们用hash表存下。这样，我们只需要算出一个表达式的优先级，即可得出它是否能够符合预期效果。
复杂度：$\mathrm{O}(100^2 * 100 * 10)$

提示与补充：hash表我们可以调用map$<string,int>$，这样会很方便。

1.2 8D Two Friends

1.2.1 Description

题意：平面上有三个点A,B,C，现在有甲乙两人共同从A出发，甲需要从A到B再到C，乙需要从A到C，现在他们两人由于某种特殊的原因（自己YY去吧），想要在一起走尽量长的时间，但是甲不能走多于最优距离t1，乙不能走多于最优距离t2，问他们在一起走的长度最长是多少？当他们分开后再汇合就不能算在一起走了= =。

数据范围：t1, t2 ≤ 100, 所有的坐标的范围在[-100,100]之间，输入都是整数。

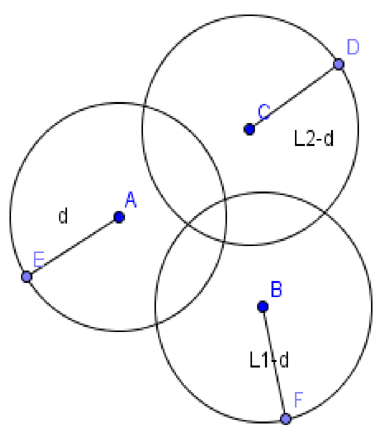
难度：★★★

1.2.2 Analysis

算法：计算几何（圆交or 三分）

分析：首先如果乙能够陪甲到B，那么最优答案便是甲和乙能走的最大距离取较小值。特判掉这种情况后，问题变成了甲与乙一起到达一个特定的点D，甲从D到B再从B到C,乙从D到C，由于B到C的距离是一定，所以我们可以直接看成甲需要从D到B即可。

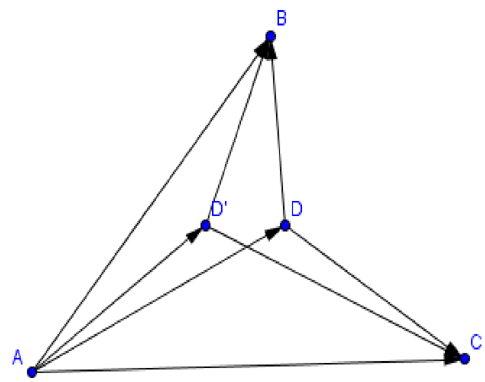
思路1：显然答案是可以二分的。问题在于如何判断一个答案是否合法。让我们定义L1 = t1 + dis(A, B), L2 = t2 + dis(A, C)。当我们二分的答案为d时，则我们需要找到一个D点，使得dis(A, D) ≤ d, dis(B, D) ≤ L1 - d, dis(C, D) ≤ L2 - d，如下图：



于是乎，问题转化成判断三个圆有没有交集。判断方法可以枚举两个圆，

求出它们的交点，再判断这个交点是否在第三个圆内。

思路2：三分。我们以A点建立极坐标系，易得DA将夹在BA和CA之间，当D的角度确定时，它的长度可以二分出来。而D的角度thita所对应的最长长度L应该是成单峰性的，你可以想象：将在峰值处取到平衡点，而越往两边走，便越不平衡了。如图所示：



提示与补充： 计算几何=小心精度。特别注意判断两个圆有没有交时，小心不要把相切的情况判断成相离。

1.3 8E Beads

1.3.1 Description

题意：问第k小的满足以下条件的N位01串是什么。它比自己取反、翻转、取反后翻转都要小或者相等。

数据范围：N≤50，K≤ 10^{16}。

难度：★★★

1.3.2 Analysis

算法：DP（按位统计的思想）

分析：考虑题目中的三个条件：

1. 取反：需要直接令第一位等于0。
2. 翻转：第一位和最后一位相同时，剩下的n-2 位需要满足小于等于翻转后的串，当第一位为0，最后一位为1时，剩下的n-2位可以任取。
3. 取反后翻转：第一位和最后一位不同时，剩下的n-2位需要满足小于等于取反后翻转的串，当第一位为0,最后一位为0，剩下的n-2位可以任取。

也就是说，这种定义实际上是一种递归的形式。设计状态f[n][0..1][0..1]，n表示长度为n，第二维表示是否需要小于等于翻转后的串，第三维表示是否需要小于等于取反后翻转的串，f[n][0..1][0..1] 表示满足条件下的方案数。再回到我们的问题，假如我们确定了这个N位串的前i位，怎么得出这种情况下的方案数呢？不难发现，我们只需要记录下，确定前i位后，在后i个任选的方案中的g[0..1][0..1]数组，g数组表示翻转，取反后翻转是否会相等的方案数。这样g数组和f数组的组合就能够得出准确的方案数。

于是我们可以按位确定最后的方案。当我们确定了$N/2$个以后，实际上这个时候的方案数最多只有两种（一种是翻转，一种是取反后翻转），特判一下就好了。

时间复杂度$O(n)$

提示与补充：无

1.4 10E Greedy Change

1.4.1 Description

题意：现在你有n种面值不同的硬币，第i种硬币面值为ai。你想要用这些硬币凑出x元钱。你将采取如下的策略，每次找到$< x$的最大面值的硬币ai，取一枚ai，再用上述策略去凑x-ai。但是很多时候这样都不一定是最优的。即不一定是用最少的硬币数凑出了x。请问令这个贪心策略失效的x最小是多少？

数据范围：$n \leq 400$
难度：★★★

1.4.2 Analysis

算法：贪心
分析：关于本题的证明有一篇严谨的英文论文：《A Polynomial-time Algorithm for the Change-Making Problem》。

下面是摘自网上对该英文的部分翻译：

一个货币系统是一个有限整数集，$c1 > c2 > .. > cn$，令cn=1，否则可能无解。

定义C为n元向量(c1,c2,..,cn)，在C下一个非负整数x可用一个满足V • C=x，非负n元向量V表示。V的大小表示为$|V|$=V • (1,1,..,1)。

定义>为字典序的大于。

G(x)是x的贪心表示。

M(x)是x的最小且字典序最大表示。

V是U的子集当且仅当对于任意$1 <= i <= n, vi <= ui$。

我们要证明存在$G(x) \neq M(x)$。

引理：

若U=G(U • C)，则对于任意V为U的子集，那么V=G(V • C)

若U=M(U • C)，则对于任意V为U的子集，那么V=M(V • C)

证明：

U=G(U • C)。

V为U的子集。

对于任意V‘ • C=V • C

V‘ • C=V • C

(U-V+V‘) • C=U • C

$(U - V + V‘) <= U$ （因为U是字典序最大的）

$V‘ <= V$ （移项）

定义运算<<，$A << B$，当且仅当$|A| > |B|$或者$(|A| = |B|$且$A < B)$。

显然运算<<满足<所满足的所有性质。

对于第二个引理证明可以仿照第一个。

对于一个货币系统，若存在最小的反例w使得$G(x)! = M(x)$，定义

$A = G(x) = (a1, .., an)B = M(x) = (b1, .., bn)$

显然满足对于任意$1 <= i <= n$，不存在$ai > 0$ 且$bi > 0$，因为若$ai > 0$且$bi > 0$，令ai和bi 同时减一，通过引理我们可以知道，依然是贪心表示和最小表示，且不相等，那就存在更小的反例w-ci，与定义矛盾。

定义M(w)=(m1,..,mn)，i为其第一个非零位，j为其最后一个非零位。

通过定义可得$M(w) < G(w)$，那么gi=0，且存在之前的某位不为0，所以$i > 1$。

定理

对于1到j-1号元素，M(w)的和G(c(i-1)-1)的相等。且M(w)的j+1到n号元素都为0。

证明：

因为G(w)第i位为0且非0位在更早的位置，那么说明$w >= ci - 1$。

通过引理可得，M(w)在j位减一，依然还是最小表示，又因为不存在更小的反例，可得G(w-cj)=M(w-cj)，那么G(w-cj)的i位之前全为0，所以
$w - cj < c(i - 1) <= w$

定义V=(v1,v2,..,vn)=G(c(i-1)-1)。

$c(i - 1) - 1 >= ci$,所以vi不等于0。所以我们可以将vi和mi同时减一来得到新的贪心表示G(c(i-1)-1-ci) 和M(w-ci)=G(w-ci)，从前面的证明中可以得知，$c(i - 1) - 1 - ci < w - ci$，所以$G(c(i - 1) - 1 - ci) < G(w - ci)$。某位同时加一不会影响字典序，所以有

$V < M(w)$

同时，我们可以将mj位减一，那么可以得到M(w-cj)=G(w-cj)。因为$w-cj <= c(i-1)-1$，所以可得$G(w-cj) <= V$。可以发现G(w-cj)和M(w)的区别只有第j位，所以如果在第j位前V与M(w)不同，那么他不可能字典序在二者中间。但是V¡M(w)，所以必须存在$k >= j$，vk¡mk。但是m(j+1)..mn都是0，所以vj¡mj。但是$G(w - cj) <= V$，所以$mj - 1 <= vj$。所以mj只能等于vj +1。

所以枚举i、j,然后验证即可。

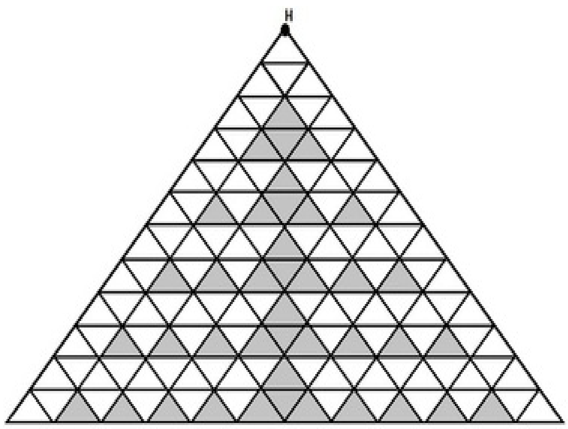
复杂度$O(n^3)$。

提示与补充： 无

1.5 15E Triangles

1.5.1 Description

题意:



一个有n层的森林，如图所示(n=12)。现在从点H出发，走某一条路径，最后又回到H点。这条路径需要满足以下几个条件:

1. 路径将形成一个简单多边形：即一个点不能被经过多次
2. 简单多边形不能包含灰色的格子，灰色格子规律可以由上图推广。

问这样的简单多边形有多少个?

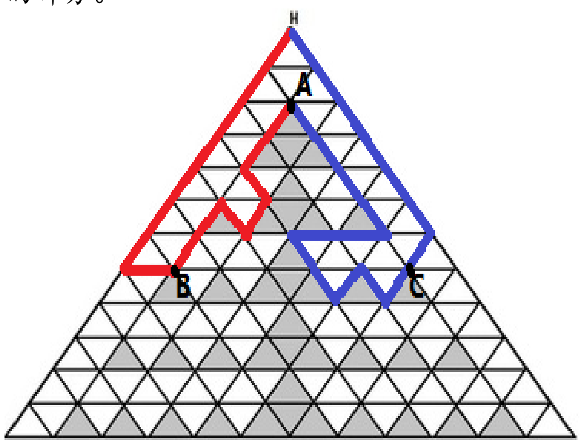
数据范围: $n \le 10^6$

难度: ★★

1.5.2 Analysis

算法： 组合分析

分析： 首先可以发现多边形能沿着H点的竖直方向切成两个互相不干扰的部分。



我们现在只考虑左边红色部分的方案数，最后再乘起来就可以了。红色部分又可以分成两部分：由H到B 和由A 到B。H到B的方案数只有两种，而A到B 的方案数取决于是否绕进去"曲折图形"：  ，跨过n个曲折图形组成的图案的方案数可以递归求得：$f[n] = 3 + 2 * f[n-1]$。所以我们枚举B点的位置，算出A到B的方案数，即可求出红色部分方案数，蓝色部分与其等价，相乘即可。最后注意有一种不经过A点的方案。

提示与补充： 本题只需分析仔细即可。

1.6 17C Balance

1.6.1 Description

题意：给定一个字符串，你每次可以将一个字符去替换它相邻的一个字符，也就是将ab变为aa 或者变为bb。已知这个字符串只包含a,b,c，并且定义一个平衡的字符串为：a,b,c三种字符出现的次数的相差值不超过1。现在问通过若干次操作总共能够得到多少个不同的平衡的字符串

数据范围：$n \leq 150$

难度：★★★

1.6.2 Analysis

算法：DP

分析：可以发现，最后的字符串将由若干个字符相同的段构成，我们写下每一段的字符，可以发现这将构成原串的子序列。于是我们定义状态F[n][a][b][c]，代表匹配到了原串的第n个（第n个一定匹配），当前的a,b,c的个数确定时的方案数。转移时只需要枚举最近的一个不相等的字符进行转移即可。应用前缀和优化可以将转移降至$O(1)$, 总复杂度$O(n^4)$。

提示与补充：注意$a, b, c \approx n/3$，所以数组不会爆空间。

1.7 17E Palisection

1.7.1 Description

题意：给定一个长度为N的字符串，问相交的回文子串有多少对？
数据范围：N$\leq 2 * 10^6$
难度：★★★

1.7.2 Analysis

算法：Manacher+数据结构
分析：首先用Manacher算法在O(n)的时间内求出r[i]数组，代表以i为中心的回文子串的最长长度。然后我们考虑计算问题的补集：不相交的回文子串有多少对。这个问题我们可以枚举分割线，令ed[i]代表以i结尾的回文子串的个数，st[i]代表以i开头的回文子串的个数，则不相交的回文子串的对数为：

$$\sum_{i=1}^{n} st[i] * \sum_{j=1}^{i-1} ed[j] \tag{1.1}$$

通过简单的差分和前缀和优化我们可以在O(n)的时间内将上述式子求出。

提示与补充：关于Manacher算法的资料可以查看这里

1.8 19E fairy

1.8.1 Description

题意：在一个N个点,M条边的无向图中，问有多少条边满足删掉它后使得整个图是一个二分图。

数据范围：$N \leq 10^4, M \leq 10^4$

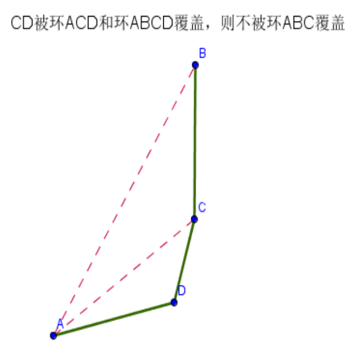
难度：★★★

1.8.2 Analysis

算法：dfs+树上的路径操作

分析：二分图的定义是整个图不存在奇环。所以我们的任务是删除一条边，使得整个图不存在奇环。一个显然的想法是找出整个图中的奇环，然后从它们的交集中任选一条。但是环的个数是指数级的，所以这个方法复杂度太高了。

考虑对整个图进行dfs，将边分成环边和树边。我们定义树上的环为一条环边加若干条树边构成。可知，图中所有的环都可以由树上的环异或得到（一条边出现两次等价于没有出现），所以我们的答案肯定在树上的奇环的交集中产生。可是如果这条边同样被一个偶环覆盖了，那么该偶环和奇环的异或环就不包含这条边，换言之，如果一条边同时被奇环和偶环覆盖，那么必然存在另一个奇环不覆盖这条边。如图所示：



所以我们需要找出被所有奇环覆盖并且不被任何一个偶环覆盖的树边。现在我们只考虑一条边被奇环覆盖的次数，设一条奇环连接着(u,v)，u是v的祖先（因为是dfs的缘故，所有的环边都是回边），令f[v]++,f[u]–,这样v节点的子树的f值之和就是v连向父亲的边被覆盖的次数（差分的原理）。所以我们只需要一遍dfs就可以求出每条边被覆盖的次数分别是多少。时间复杂度O(n)。

提示与补充：注意没有奇环和树上只有一个奇环的情况。

1.9 23D Tetragon

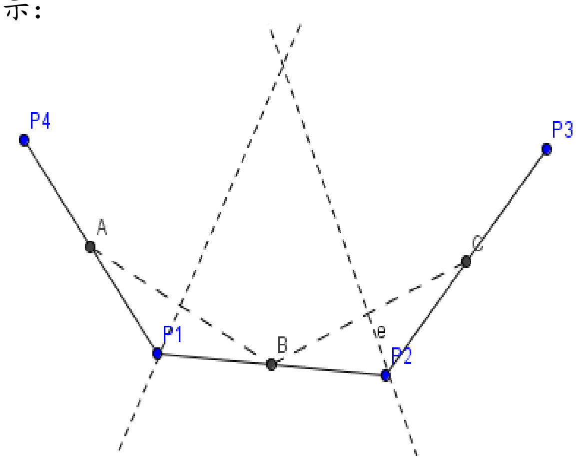
1.9.1 Description

题意：给定一个凸四边形的三条相等边的中点，请还原这个凸四边形。
数据范围：$T \leq 5 * 10^4$
难度：★★

1.9.2 Analysis

算法：计算几何+线性代数（解方程）
分析：我们假设三个中点按顺时钟或者逆时钟顺序为A,B,C，则我们需要找到两个关于B对称的点为P1,P2，使得$\begin{cases} dis(B,P1) = dis(A,P1) \\ dis(B,P2) = dis(C,P2) \end{cases}$

如图示：



易得P1点落在A,B的垂直平分线上，P2落在B,C的垂直平分线上。于是我们尝试解方程的思想解决这个问题。我们令A,B的垂直平分线表示成O1+G1*x（点+方向*长度），C,D的垂直平分线为O2+G2*y。得:

$$\begin{cases} P1 = O1 + G1 * x \\ P2 = B + (B - P1) \\ (P2 - O2) \times G2 = 0 \end{cases} \tag{1.2}$$

化简此式可以解出x，于是在O(1)的时间可以得出其它顶点再进行凸性检测。另外此题我感觉三分也是可行的=，但是肯定没有直接解方程来的优美。

提示与补充：当两条垂直平分线平行时，直接判无解就可以了，以免解方程时0作为除数。

1.10 23E Tree

1.10.1 Description

题意： 给定一棵N个点的树，请你删去某些边，使得剩下的各个联通块大小的积尽可能大。

数据范围： $N \leq 700$

难度： ★★

1.10.2 Analysis

算法： 树形DP

分析： 引理：存在一种最优分块中使得不存在一条长度等于3的路径

证明：我们断开中间那一条边，设两边的联通块大小分别是a和b，$a \geq 2, b \geq 2$，则$ab \geq a + b$，所以断开一定不会变差。

于是我们可以设计如下的一个DP算法。h[u]代表以u为根的子树的最优值，f[u]代表不考虑根时子树的最优值，g[u][s]代表u和它的s个儿子在一个联通块时，整棵子树的最优值（不考虑u所在联通块）。对于u来说，就它的子树而言，它所在的联通块只有三种情况：

1. u单独在一个联通块
2. u和若干个儿子在一个联通块
3. u和某个儿子v,以及v的若干个儿子在一个联通块中

所以DP方程也很简单(复杂度为$O(n^2)$[不考虑高精度])：

$$f[u] = \prod h[v] \text{ (遍历u的儿子v)} \quad (1.3)$$

$$g[u][s] = max(g[u][s-1] * f[v], g[u][s] * h[v]) \text{ (遍历u的儿子v)} \quad (1.4)$$

$$h[u] = max(f[u], g[u][s] * (s+1), g[v][s] * (s+2)\text{[遍历u的儿子v]}) \quad (1.5)$$

提示与补充： 其实此题用状态f[u][s]表示u 为根的子树，u所在联通块有s个点时子树的最优值，也能通过。这个状态转移的时候类似于背包，看似复杂度n^3，但其实远远达不到，我猜测最坏情况是$n^2 \log n$[1]，在codeforces 上能够很快的通过全部数据。

[1]如果你有不同的想法或者有关于这个复杂度的证明，欢迎发邮件和我交流：380759124@qq.com。

Chapter 2

codeforces(2)

2.1 26E Multithreading

2.1.1 Description

题意：现在有n个如下的进程：

```
repeat n_i times
    y_i := y
    y := y_i+1
end repeat
```

其中y是一个公共的变量，yi是每个进程特有的变量，一开始所有的y和yi都是0。如果按顺序调用这n个进程，那么最后的结果将是y = $\sum ni$，其中第i个进程被调用了2*n[i] 次。可是如果将调用的顺序打乱的话，最后的y就可能不一样。比如n[1] = 2, n[2] = 1，我们按如下的形式调用进程：2, 1, 1, 1, 1, 2，最后的y就会是1。
现在给定n[i]和w,问最后y是否能变为w。

数据范围：$n \leq 100, n[i] \leq 1000, -10^9 \leq w \leq 10^9$
难度：★★

2.1.2 Analysis

算法：贪心构造
分析：当n = 1时，直接判断即可。接下来我们约定ni为递增序列。

当$w > \sum ni$ 或者$w < 0$时，我们可以直接输出无解。

当$n1 \leq w \leq sum$时：

首先进行(n1-1)*2次第一个进程，这时候y = n1 - 1，然后接下来按顺序运行其它进程，当y = w - 1 时，运行一次1进程，使得y1 = w - 1，最后要结束时再运行一次1进程，y = y1 + 1 = w。

当$2 \leq w \leq n1 - 1$时：

首先进行一次2进程，然后进行(n1-1)*2次1进程，然后进行(w-1)*2-1次2进程，此时y = w - 1，再运行一次1进程，y1 = w - 1,运行完其它进程后最后运行一次1进程，y = w。

提示与补充：注意输出方案

2.2 28D Don't fear

2.2.1 Description

题意：从1到n有n个元素，每个元素有四个域：v[i],l[i],r[i],c[i]。现在要求删除某些元素，使得剩下的元素满足：
如果元素i没有被删除：

l[i]= $\sum$\{c[j] $|j < i$ 且第j个元素没被删去\}

r[i]= $\sum$\{c[j] $|j > i$ 且第j个元素没被删去\}

在满足这些条件前提下，要求$\sum$\{v[i] | i没被删去\} 最大，请输出这个最大值，并且依次输出保留下来的元素的编号。

数据范围：$n \leq 10^5$

难度：★

2.2.2 Analysis

算法：DP

分析：我们假设在最优方案中有着两个相邻的元素j,i($i > j$)。则他们需要满足的条件是：

l[i] = l[j] + c[j];

r[j] = r[i] + c[i];

定义F[i]表示保留第i个元素，前i个元素能够得到的最优值。
状态转移：F[i] = max(F[j]) + v[i]\{j和i满足上述条件\}。所以我们用一个set记录下(l[j]+c[j], r[j])的最优值，转移时查询(l[i], r[i]+c[i])即可。
算法复杂度O(nlogn)

提示与补充：无

2.3 30D King's Problem

2.3.1 Description

题意：平面上有n+1个点，前n个点的坐标分别是(x[1], 0), (x[2], 0) ... (x[n], 0)。第n+1个点的坐标为(x[n+1], y[n+1])，现在要求从第k个点出发，遍历所有的点，问最短的长度是多少？

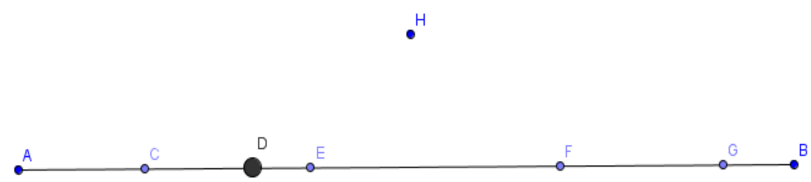
数据范围：$n \leq 10^5$

难度：★★

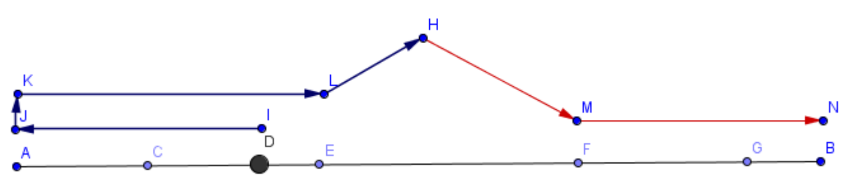
2.3.2 Analysis

算法：贪心

分析：假设$k \neq n+1$



如果下图是一种最优方案：



将路线分成两部分，一部分是在到达n+1点之前的蓝色路线，一部分是到达n+1点之后的红色路线。我们考虑第一次到达n+1点后的情形，这时候数轴上的点我们只需要考虑还没有被覆盖的最左端和最右端即可。而在出发前往n+1点之前，数轴上必然覆盖的是最左端的若干个点或者最右端的若干个点。所以我们枚举这若干个点的个数即可。更简单来说：我们需要枚举一个断点i，使得k覆盖完[1,i]（或者[i,n]）后出发去n+1,再令n+1回来覆盖[i+1,n]（或者[1,i-1]）。

当$k = n + 1$时，直接令k回来覆盖[1,n]即可。

提示与补充：注意i=n的情形，这时候n+1不需要再回来了。

2.4 30E Password

2.4.1 Description

题意：现给定一种对奇长度回文串加密的方法：将串按顺序划分成连续的ABC三段，其中A和C 段长度必须相等，另有DEF三个字符串，则加密后的串为DAEBFC。先给定加密后的串，求原串最长是多少，要求方案。

数据范围：串长$\leq 10^5$

难度：★★★

2.4.2 Analysis

算法：数据结构+字符串处理

分析：枚举C的长度，我们需要得到一个能够匹配C的A，并且A的位置尽可能靠前。将加密串倒序做一次KMP即可胜任此任务。假设C的开头位置是r,A的结束位置是l，我们需要查询一个在(l,r)区间的回文子串，并且子串的长度尽可能长。

首先用manacher算法求出r[i],r[i]代表以i为中心的最长回文子串的长度。然后可以用各种数据结构对其进行分类讨论+优化即可。复杂度$O(nlogn)$。

这里着重介绍一种更好的O(n)算法。我们不再枚举C的长度，而是改为枚举B的中心，因为一定存在一种最优方案使得B是极长的回文子串。这时我们预处理出一个f[i],代表A在1-i中，C最长是多少。枚举B后得到极长的回文串:[i-r[i], i+r[i]]，如果这时候C的最优长度覆盖了B，只需要让C 退回去即可。所以整个算法将在O(n)的时间内完成。

提示与补充：无

2.5 32E Hide-and-Seek

2.5.1 Description

题意：平面上给定两个点A,B，一面能够反射光线的镜子(Ms,Mt)，还有一面墙(Ws,Wt)，现在问A是否能够看到B（可以通过镜子）。如果视线被墙挡住，就看不见，如果视线被镜子挡住，就会发生发射。

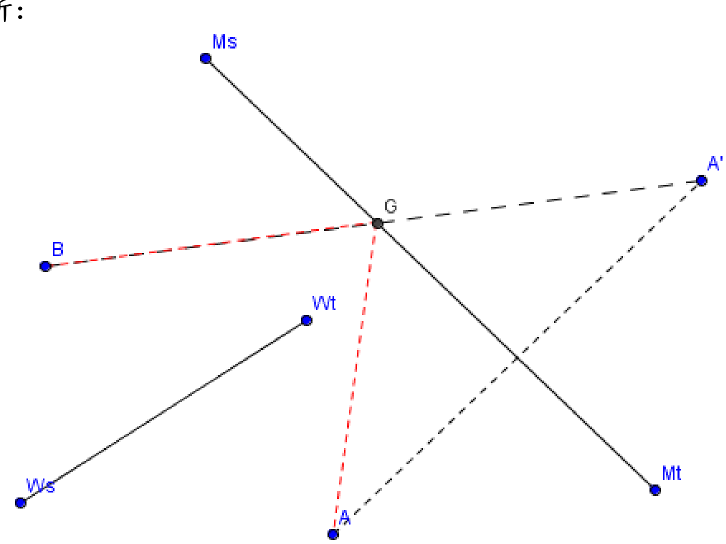
数据范围：无

难度：★★

2.5.2 Analysis

算法：计算几何

分析：



我们首先判断A,B可以直接看到的情况，然后是在镜子两侧的情况，最后判断需要反射的情形。做A关于镜子的对称点A’，连接A’和B，交镜子于G点，然后就是判断AG和BG分别会不会被墙挡住即可。

提示与补充：注意共线的情况，建议提前处理掉。

2.6 35E Parade

2.6.1 Description

题意：给定N个下边界紧贴X轴，上边界在X轴上方的矩形（矩形的四边都平行于坐标轴）。求这个矩形的“轮廓”，即输出矩形的并的顶点坐标，要求轮廓相邻的边垂直。

数据范围：$N <= 10^5$

难度：★★

2.6.2 Analysis

算法：数据结构

分析：直观上来看，我们每次沿这边界行走，就能得到最后的答案，但是这样处理起来比较复杂。用另外一种更为简单的思路，我们将最后轮廓中所有的横向线段找出来，再把他们连接起来即可。考虑某个横坐标所对应的轮廓的纵坐标，实际上是覆盖它的所有横向线段中最高的那条。我们将所有的端点离散出来（因为转折点的横坐标只会在端点处产生），然后按横坐标从小到大依次扫描每个端点，同时维护当前的轮廓的纵坐标，如果纵坐标发生了变化，则我们需要"拐弯"，将这个转折点记录下来即可。维护当前轮廓的纵坐标可以用系统堆。复杂度$O(nlogn)$。

提示与补充：无。

2.7 36E Two Path

2.7.1 Description

题意： n个点m条边的无向图，请找出两条路径，使得每条边被经过且只经过1次。

数据范围： $n, m \leq 10^4$

难度： ★★

2.7.2 Analysis

算法： 欧拉路径

分析： 如果只需要一条路径经过所有的边，那么便是非常经典的欧拉路径问题。如果一个图满足如下两个条件:

1. 整个图联通
2. 奇度点的个数为0或者2

那么这个图就存在一条欧拉路径覆盖所有的边恰好一次。具体做法为dfs所有的边，取它的反出栈顺序。

如果扩展到要两条路径呢？那么我们需要处理的就是四个奇度点的情况。设这四个点分别u1,u2,v1,v2，这里介绍一种颇为巧妙的方式。我们首先新建一条边连接u2 和v1，设该边为P,然后整个图只剩下两个奇度点:u1,v2，我们求出该图的欧拉路径，设其为e1,e2...P,el,el+1...em。接下来只需要把这条路径由P 切割成两个部分，即可得到满足要求的两条路径！

提示与补充： 注意处理多个联通块的情况。

2.8 37E Trial for Chief

2.8.1 Description

题意：现在有一个n*m的初始全是白色格子的矩阵，每次可以选择一个四联通块将它染成黑色或者白色，问最少染多少次能够得到目标矩阵。

数据范围：$n, m \leq 50$

难度：★★★

2.8.2 Analysis

算法：贪心+图论

分析：此题在钟沛林同学的集训队互测题中已经出现过，下面的文字摘自他的题解：

首先我们建立这样一张无向图：对于每个格子我们视为一个顶点，在相邻的四个格子中，如果在目标图里它们的颜色一样，就连一条边权为0的边，若颜色不同就连一条边权为1的边。现在我们对于每个格子，以它为起点计算到其他点的最短路，我们找到离这个格子最远的黑色格子，若这个距离是D，我们容易用贪心的思想构造一个长度为D+1 的染色序列来达到目标：

1. 第一步把所有距离¡=D的格子染黑。
2. 第二步把所有距离¡=D-1的格子染白。
3. ……
4. 最后一步：把所有距离¡=0 的格子染成对应颜色。

这样，我们只需要在所有格子中选一个以它作为起点，最远黑色格子距离最短的格子，那么它的D+1 就是答案。这样做的合法性是显然的，我们来证明这样做的最优性。首先我们把所有距离为0的点压缩成一个点。注意如果我们如果给某些格子染色了的话，我们把这些格子距离为0的点一起染了结果不会有影响。

1. 我们可以证明任意方式的染色，总可以转变这样的形式：对于每一个染色操作所操作的连通块，它总是前一次操作的连通块的子集。这个是可以比较意识流的感觉出来的，多画几次就明白了。
2. 最后一次操作恰好只要染一个同色连通块。

因此，存在一个最优解是如上方式操作的。时间复杂度：$O(n^2m^2)$

提示与补充：注意这是一个平面图，边数与点数同级。

2.9 39C Moon Craters

2.9.1 Description

题意： 数轴上有n条线段，要求找出尽可能多的线段，使得这些线段中不存在两条线段部分相交（内含不算部分相交）。请输出方案，并保证不存在两条线段完全相等。

数据范围： $n \leq 2000$

难度： ★★

2.9.2 Analysis

算法： DP

分析： 先将将所有的端点离散。定义F[l][r] 代表l到r这一段中最多能取多少条线段。

$$F[l][r] = \begin{cases} F[l][r-1] & (\text{如果不取r为右端点的线段}) \\ F[l][t] + F[t][r-1] + 1 & (\text{t为以r为右端点的左端点编号，考虑取r的情形}) \end{cases}$$

按如上方程转移即可，复杂度$O(n^2)$

给端点排序时需要注意：右端点应排在左端点前，同为右端点时，左端点靠后的排在前面，同为左端点时，右端点靠后的排在前面。

提示与补充： 此题有更为高效的$O(nlogn)$ 做法，希望有兴趣的同学能够继续思考。

2.10 39A Calculations

2.10.1 Description

题意： C*++语言是这样定义的：

1. 表达式=基本式or 表达式+基本式or 表达式-基本式
2. 基本式=增量or 系数*增量
3. 增量=a++ or ++a
4. 系数=0/1/2/……/1000

如“5*a++-3*++a+a++”是合法的C*++表达式。计算这样的表达式的值的方法：首先是每个基本式进行计算，然后按照正常的算术运算法则计算。如果一个基本式包含“a++”，则先进行乘法运算再使变量a权值+1；如果一个基本式包含“++a”，则先使变量a 权值+1再进行乘法运算。然而基本式可以按任意顺序计算，这就是为什么计算结果是完全无法预料的。现在给定一个表达式和a的初始值，你的任务就是去找到最大的可能结果。

数据范围： 表达式长度$\leq 10^4$,系数的绝对值$\leq 10^3$,a的初始值$\leq 10^3$
难度： ★(少有的1星题哦)

2.10.2 Analysis

算法： 贪心
分析： 我们将所有基本式都预处理出来，此题的目标就是要给基本式安排一个运算顺序，使得最后的结果最大。每个基本式的形式是一个系数ki*x，我们考虑两个相邻的基本式系数为k1,k2，可以用调整法证明系数小的放前面一定更优。所以我们只需要将ki从小到大排序，然后依次计算即可。
提示与补充： 无。

Chapter 3

codeforces(3)

3.1 39E Dirichlet

3.1.1 Description

题意： 给定a,b,n，两个人轮流操作，每次可以令a+1，或者b+1,使$a^b \geq n$ 的人输，问先手是否有必胜策略，或者打成平手。

数据范围： $1 \leq a \leq 10^4, 1 \leq b \leq 30, 2 \leq n \leq 10^9, a^b < n$

难度： ★(难得一见的1星题=)

3.1.2 Analysis

算法： 记忆化搜索

分析： 用f(a,b)代表先手面临(a,b)状态时是否能够必胜，0代表必败，1代表必胜，2代表平局。显然f(a,b)可以由f(a+1,b)和f(a,b+1) 得到。问题在于状态的个数在什么级别?当$a \geq 2, b \leq 31$，当$b >= 2, a <= sqrt(n)$,所以我们只需要特殊处理$a = 1 or b = 1$ 的情形。

1. 当a=1,且$(2^b) \geq n$时,两人会选择不断令b+1,所以是平局。

2. 当b=1,且$(a^2) \geq n$时，两人会选择令a+1，此时判断a和n的奇偶性即可。

复杂度O(sqrt(n))

提示与补充： 无

3.2 40E Number Table

3.2.1 Description

题意： 在一个n*m的矩阵中，有k个格子已经确定了是0或1，请确定其它格子的取值，使得每行每列的异或和都为1。问方案数是多少？

数据范围： $n, m \le 10^3, k < max(n, m)$

难度： ★★

3.2.2 Analysis

算法： 贪心

分析： 本题有一个很特殊的性质：$k < max(n, m)$，我们应该从这个突破口入手。这个性质保证了一定存在一行或者一列：该行（列）上没有一个格子被确定。我们让该行最后被安排，由于前面n-1 行都已经确定，所以该行的每个格子实际上也已经被确定了。注意到由于n行的异或和是1，m列的异或和也是1，所以n和m必须是同奇偶的，否则问题将无解。在判断这种情况后，我们只需要让前n-1行任取，最后一行一定能够满足题目的要求。而所谓任取，只需要保证该行的异或和为1即可。如果该行有cnt个格子被确定，那么该行的方案数便是$2^{m-cnt-1}$，注意要特判该行被全部填满的情况。

下图是一个例子（黑色为已经确定，蓝色为被前面n-1行确定）

1	1	0	1
1	0	1	1
0	0	1	0
1	0	1	1

提示与补充： 无

3.3 43E Race

3.3.1 Description

题意：现在Berland正在举行一场赛车比赛。比赛赛道可以看成一条s公里长的直线。有n辆赛车参加了这个比赛，他们同时从赛道的起点出发。我们已经知道每辆赛车的行为——在每一阶段中赛车的速度是恒定的。第i辆赛车的第j个阶段可以用数对(v(i,j),t(i,j))表示，v(i,j)表示第i辆赛车在第j个阶段的速度，单位是公里每小时，t(i,j)表示第i辆赛车在第j个阶段的行驶时间，单位是小时。每个阶段按顺序给出，赛车将按照这些行驶。

你的任务就是统计这次比赛中出现了多少次超车，超车指一辆车行驶到了另一辆车的前面。所有超车都是在一瞬间完成的，也就是说不会存在两辆车在一段正长度的时间中并行的情况。可能存在同一时间发生多次超车的情况，此时每一个超车都必须被统计。两辆车在赛道的起点或终点相遇的情况不需要统计。

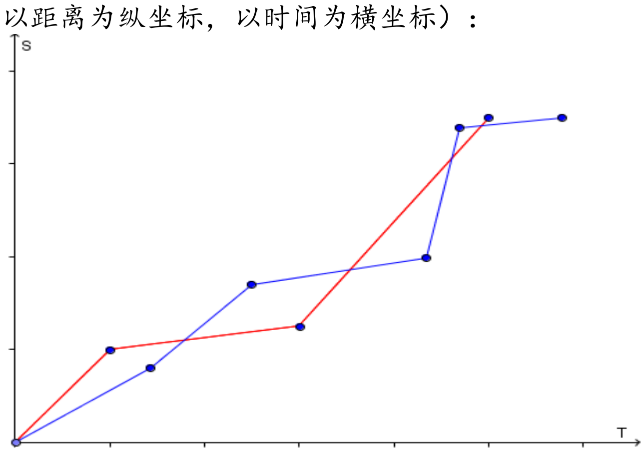
数据范围：$n \leq 100$,每辆车的阶段数≤ 100

难度：★★

3.3.2 Analysis

算法：暴力

分析：枚举两辆车，考虑它们的超车情况。将它们的行驶情况绘制如下（以距离为纵坐标，以时间为横坐标）：



由图可知，我们的目标是要统计这两条折线段的交点的个数。我们可以采取如下的方式：将每个端点处两辆车的前后关系处理出来，然后判断两个相邻的端点是否发生了超车。值得注意的是某些端点可能存在两辆车重合的情况，这时候我们需要判断前一个端点和后一个端点的前后关系。

提示与补充：无。

3.4 44J Triminoes

3.4.1 Description

题意：给定一个n*m的矩阵，某些格子为障碍，剩下的格子染上了黑色或者白色，现在要求用一个”白黑白”的1*3的骨牌去覆盖所有被染色的格子（可旋转），请输出一种方案。

数据范围：$n \leq 1000, m \leq 1000$

难度：★★

3.4.2 Analysis

算法：贪心

分析：我们考虑第一排，从左往右，如果出现白色，那么必须要通过一个横向的骨牌覆盖。也就是说如果出现黑色，我们应该先贪心横向的放，如果放不了，就必须竖向的放。按照这样的规则贪心，最后就能得出一组合法的解（实际上解也是唯一的）。

例如下图：

```
.w.wbw.wbw  .a.aaa.ccc
wbwbw.w.w.  baccc.c.a.
bw.wbwbwbw  ba.dddcbab
w.wbw.wbwb  b.aaa.cbab
...wbw.w.w  ...bbb.b.b
..wbw.wbw.  ..ccc.ddd.
```

复杂度O(n*m)。

提示与补充：骨牌可以只用四种颜色表示出来（四色定理）。

3.5 45G Prime

3.5.1 Description

题意：将1-n分成尽可能少的集合，使得每个集合的元素的和为素数。
数据范围：$n \leq 6000$
难度：★★

3.5.2 Analysis

算法：数论
分析：哥德巴赫猜想：任意一个大于3的偶数，都可以表示成两个素数的和（1+1 = 2）。

这是数学皇冠上的"明珠"，为了证明该猜想人类已经努力了几百年。现在通过计算机已经在较小范围内验证了该猜想的正确性，而我们这题也需要借一借哥德巴赫猜想的"光芒"：

$$\begin{cases} \text{n是偶数时，枚举小于它的素数p，判断n-p是不是素数} \\ \text{n是奇数时} \begin{cases} \text{n是否是素数} \\ \text{n-2是否是素数} \\ \text{将n-3表示成两个素数之和} \end{cases} \end{cases} \tag{3.1}$$

提示与补充：从测试结果来看，将一个偶数分成两个素数时，有一个素数会比较小，所以能够通过全部的数据。

3.6 45E Director

3.6.1 Description

题意：现在有n个姓氏，n个名字，要将它们一一配对起来，按照如下格式输出："姓氏1 名字1, 姓氏2 名字2,...姓氏n, 名字n"。要求在满足配对的姓氏和名字的首字母相同的对数尽可能多的情况下，使输出的字符串的字典序最小。

数据范围：$n \leq 100$，名字和姓氏长度≤ 10

难度：★★

3.6.2 Analysis

算法：贪心

分析：首先，姓氏的输出显然要按照字典序的大小来，然后我们得选择每个姓氏所对应的名字。定义cnt[i]表示i首字母当前剩下的姓氏个数，size[i]表示i首字母当前剩下的名字个数，f[i]表示i首字母当前剩下的名字中字典序最小的。设我们现在要帮助首字母为t的姓氏选择名字。

1. 若cnt[t]≤size[t]，则选择f[t]。
2. 否则，从小到大选择第一个cnt[i]<size[i]的f[i]，再和f[t]进行比较。

这样的贪心的选择即可以解决该问题，具体维护时可以用set。复杂度O(nlogn*10)。

提示与补充：贪心时考虑地严谨一些。

3.7 46F Hercule Poirot

3.7.1 Description

题意：现在有n个房间，m扇门，房间与房间之间由门相连，每个门由特定的钥匙打开，一种钥匙只有一个。现在有K个居民在某些房间中，每个居民手中有一些钥匙，接下来他们可以互相走动，他们的操作包括：

1.如果该居民有对应的钥匙，打开或者关闭他所在房间到隔壁房间的门（每扇门可以从两边打开或关闭）；

2.如果门是开着的，从一个房间走到相邻的房间；

3.将任意数量的钥匙交给在同一个房间里的其他居民；

现在给定两个时刻的居民状态描述，问前一个时刻的居民能够通过走动得到后一个时刻的状态。

数据范围：$n, m, k \leq 1000$

难度：★★★

3.7.2 Analysis

算法：并查集

分析：可以发现，一扇门一旦被打开，该门连接的两个房间可以视为一个房间。所以我们用并查集将这两个房间并成一个集合。对两个时刻同时做并查集的操作，最后得到的集合信息应该完全一样（这个集合的信息包括：居民和钥匙）。

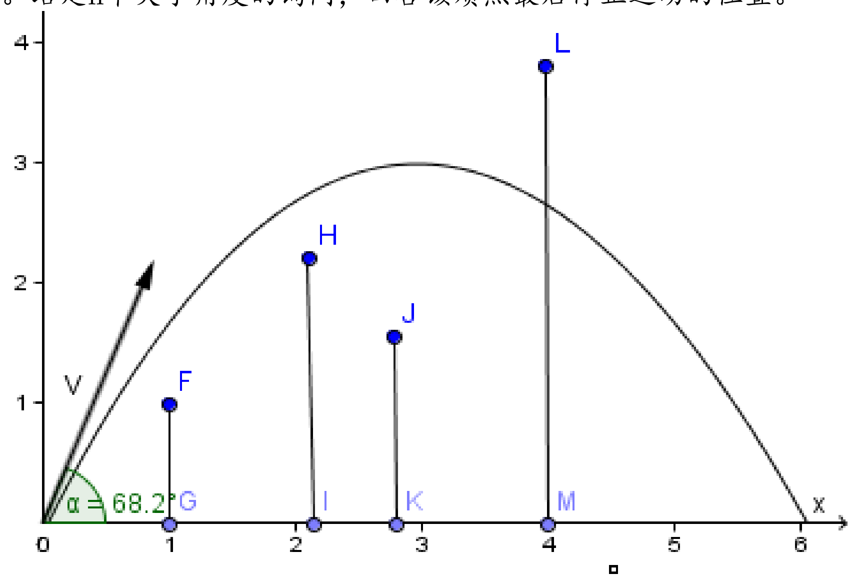
具体操作时，我们外循环m次，每次扫描每条边，检查是否能够合并，合并时将钥匙也并在一起。总复杂度$O(10^6)$。

提示与补充：对两个时刻分别做合并可以用一个函数完成。

3.8 47E cannon

3.8.1 Description

题意： 在第一象限中，我们从原点沿α角度抛出一个单位质量的质点，只受重力作用。如果不受其他因素影响，它的运动轨迹将是一条抛物线。现在有m条线段竖立在x坐标轴上，如果质点撞在了线段上，就会停止运动。给定n个关于角度的询问，回答该质点最后停止运动的位置。



数据范围： $n \leq 10^4, m \leq 10^5$
难度： ★★

3.8.2 Analysis

算法： 计算几何，二分
分析： 我们只需要找出每条线段能够阻挡的最大角度即可。由于当$\alpha < \pi/4$时，随着α的增大，整条抛物线都会增高，所以我们可以二分出这个最大角度。预处理出最大角度后，我们可以利用单调性二分的判断每个质点会在什么位置停止运动。
提示与补充： 关于本题所需要的物理知识可以去翻任意一本中学物理教材= 。。

3.9 49E Ancestor

3.9.1 Description

题意： 给定两个字符串s1,s2。定义字符串s的变异是某个字符ai变为相邻的两个字符bi,ci。如果s变异多次后得到t,就称s为t的祖先。给定变异的规则表，问s1,s2的公共祖先的最小长度。

数据范围： 字符串长度≤ 50，变异规则≤ 50

难度： ★★

3.9.2 Analysis

算法： DP

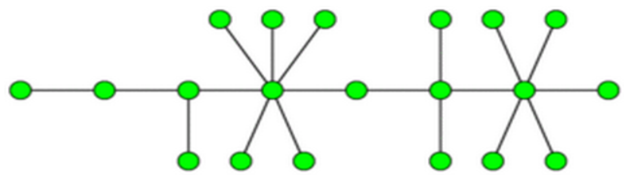
分析： 定义f[l][r][k]表示字符串s的l-r这一段能否由字符k变异到。转移很简单。分别求出s1,s2的f数组后，再定义g[i][j]。表示s1 的前i个和s2前j个的公共祖先的最短长度，转移的时候枚举下一个相同的字符即可。总复杂度$O(50^4)$。

提示与补充： 无

3.10 51F Caterpillar

3.10.1 Description

题意：定义毛毛虫图为一个无向无环图，且满足图中存在一条路径p,使得每个点到p的距离都小于等于1。如图所示：



毛毛虫图不能有重边，但是可以有自环。现在给定一个无向图，并且定义合并操作如下：每次选定两个点(u,v)，将u,v两点，以及所连边删去，再加入一个新点w,将原来和u,v相连的边改成和w相连。问至少需要多少次合并操作才能使无向图变成一个毛毛虫图。

数据范围：$n \leq 2000, m \leq 100000$

难度：★★★

3.10.2 Analysis

算法：贪心

分析：性质1：无向图中所有的环都将被合并成一个点。

说明：首先环本身是不合法的，如果我们合并其中某两个u,v，则会形成两个新的环，根据归纳法可知所有的环会被合并成一个新的点。

性质2：在一棵树中，如果我们合并(u,v)，则u到v的路径上的所有点都将被合并成一个点。

说明：因为合并(u,v)后，该路径会变成一个环，根据性质1可得性质2。由该性质可得，我们每次合并将合并两个相邻的点。

性质3：归纳证明：一棵树最后的毛毛虫图的点数为：度数为1的点数+最长链的长度-2。

说明：如果整个图只有度数为1的点和最长链，该图本身便是毛毛虫图，结论显然成立。否则我们可以合并任意一条边，由归纳可得，合并的边不影响度数为1的点数和最长链的长度时最优，于是归纳成立。

通过这3个性质，我们的思路就非常清晰了：首先缩点，然后找最长链。复杂度$O(n+m)$。

提示与补充：注意原图可能不连通

Chapter 4

codeforces(4)

4.1 53E E.Dead Ends

4.1.1 Description

题意： 给定一个n个点,m条边的无向图，求度数为1的节点为k个的生成树的个数。

数据范围： $n \leq 10, m \leq n*(n-1)/2$

难度： ★★

4.1.2 Analysis

算法： DP

分析： 令f[x][y]表示子图x中y的所有为1的位均为叶子节点的方案数，答案即所有f[$2^N - 1$][p]之和，其中p为有K个1位的数。转移时，考虑加入某个叶子节点连出的还尚未加入子图的点的某个子集，此时该叶子节点就不再是叶子节点，而加入的点将成为新的叶子节点。为了避免重复，每次选出的叶子节点应是二进制位最低的一位。具体实现可以直接按转移方程搜索，注意特殊处理从1个点转移到2个点的情况。复杂度$O(4^n)$。

提示与补充： 无

4.2 57D D.Journey

4.2.1 Description

题意： 在一个N*M的矩阵内有一些坏点，保证每行每列最多只有一个坏点，而且没有任意两个坏点对角相邻。现随机选择两个非坏点的格子分别作为起点和重点，求不经过坏点的最短路径的期望长度。

数据范围： $n, m \leq 1000$
难度： ★★

4.2.2 Analysis

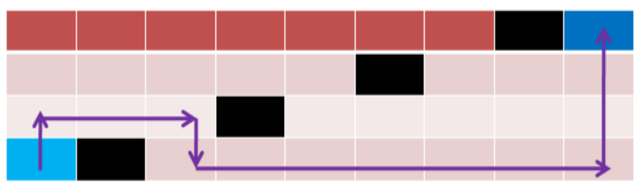
算法： 贪心
分析： 可以得出以下三个结论：

1. 任意两个格子间都是可达的

2. 两个格子的距离要么是其曼哈顿距离，要么是其曼哈顿距离+2

3. 距离为曼哈顿距离+2的格子对只有两种：
   (1)在同一行或同一列，且被坏点隔开；
   (2)两个格子之间的每一行都有一个坏点，且坏点的高度单调

如下图所示：



如果不考虑由于“绕道”而增加的距离，那么从一个点到其他所有点的距离和是可以用公式直接计算的，答案也可以用容斥原理算出来。而需要“绕道”的点对数也可以在O(N^2)的时间里统计出来。答案即两部分的和。

提示与补充： 该题的数据范围可以扩大到$n, m \leq 10^6$，希望有兴趣的同学能够继续思考。

4.3 217D Bitonix

4.3.1 Description

题意：给定T个数的集合S，现要选出S的一个子集P，使得P中的任意一些数相加或相减都得不到M的倍数，求选择的方案数。

数据范围：$M \leq 120, T \leq 10000$

难度：★★

4.3.2 Analysis

算法：DP

分析：

性质1：对于所有模M意义下相等，或者和在模M意义下为0的数，都是等价的，而且只能选一个。

性质2：子集P的大小最多为6。

证明：因为为子集P的大小为7的时候有128种和的组合，如果出现modM相等的就不合法了，而根据抽屉原理，128≥M，一定会出现不合法的情况。

于是我们可以用状压DP 来解决这道题。首先将modM相等的数分为同一类。令f[i][j]表示前i类数，j为M 位二进制，第x位代表能否凑出数x。转移时考虑是否加入当前数，用位运算进行转移。由于M较大，可以用bitset或者自定义的int128来表示j，而DP过程可以用搜索来实现。

提示与补充：bitsit是系统自带的数据结构，有关它的详细信息可以wiki。

4.4 67E Save the City

4.4.1 Description

题意：给定一个简单多边形，第一条边一定平行于坐标轴，问第一条边上有多少个整点可以看到多边形上所有的顶点（看到的定义为连线段不会落在多边形外部）。

数据范围：$n \leq 1000, xi, yi \leq 10^6$

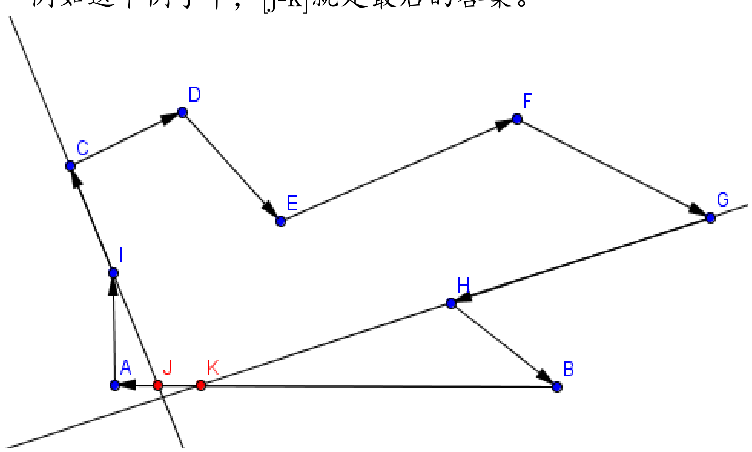
难度：★★

4.4.2 Analysis

算法：计算几何

分析：我们规定以顺时针遍历所有的边，对于某条边(a,b)，它所在的直线的左边的点一定不能同时看到点a和b。所以通过这个我们可以在第一条边上加一个限制：即只有在(a,b)所在直线的右边的点才是合法。最后我们对所有的限制取一个交即可。复杂度$O(nlogn)$。

例如这个例子中，[j-k]就是最后的答案。



提示与补充：如果不想求交点，可以枚举第一条边上的点，维护合法的左右边界，在$O(10^6)$内完成。

4.5 67C Sequence of Balls

4.5.1 Description

题意：给定两个序列A,B，现在有4个操作：

1. 在任意一个位置增加任意字母。
2. 删掉任意一个位置的字母。
3. 将某个字母用任意一个其他的字母取代。
4. 交换相邻两个字母。

每个操作都有一个花费，依次记为ti, td, tr, te，给定花费和A,B。求通过这4个操作，把A变成B的最小花费是多少。

数据范围：$2 * te \geq ti + td, len(A), len(B) \leq 4000$
难度：★★★

4.5.2 Analysis

算法：DP
分析：这题可以使用dp解决，设定状态f[i][j]表示A串匹配了i个，B串匹配了j个时的最优值
对于1,2,3操作，可以写出如下的方程

$$f[i][j] = min(f[i-1][j] + td, f[i][j-1] + ti)$$

$$f[i][j] = min(f[i][j], f[i-1][j-1] + (A[i-1] == B[j-1])?0 : tr)$$

对于4操作，可以先预处理出两个数组ga[i][alpha],gb[i][alpha]，ga表示在A中距离第i个字母最近的一个字母alpha，gb同理。又由于数据保证$2 * te \geq ti + td$。所以我们不会连续做两次交换。于是肯定是删除和添加若干元素后再交换，对于一个优的交换，它一定会把交换后的两个值匹配上，于是

令x = ga[i - 1][b[j - 1] - 'a'], y = gb[j - 1][a[i - 1] - 'a']

则f[i][j] = min(f[i][j], f[x - 1][y - 1] + te + td * (i - x - 1) + ti * (j - y - 1))

表示对于一个优的交换x、y，我们需要把A串夹在x和y中间的删除，然后交换后重新插入一些元素使得它们匹配。综合上面的转移方程就可以完整地完成转移了。答案就是f[lenA][lenB]。
提示与补充：无

4.6 70D Professor's task

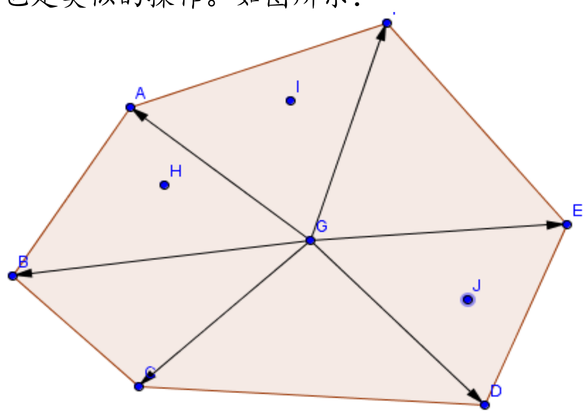
4.6.1 Description

题意：经典的动态凸包问题，包含两个操作：
1、加入一个点。2、询问一个点在不在当前点集的凸包内。
数据范围：$Q \leq 10^5$
难度：★★

4.6.2 Analysis

算法：计算几何+数据结构

分析：我们找到一个一定在凸包内的点，以它为原点建立极坐标系。可以用set找到每次插入的点所应插入的位置，然后维护整个凸包。询问的时候也是类似的操作。如图所示：



提示与补充：无

4.7 70E Reform

4.7.1 Description

题意：在一棵n个节点的树中，安排某些点为“中心点”。安排一个点为“中心点”的花费为k。某个点如果不是中心点，则它会选择最近的中心点，假设距离为len，则对答案的贡献为d[len]（d为给定的数组，保证d递增），请安排一种方案，使得总花费最小。

数据范围：$n \leq 180$

难度：★★

4.7.2 Analysis

算法：树形DP

分析：该题的DP方法比较经典。我们定义f[u][c]代表以u为根的子树，最后u选择c为”中心点”的该棵子树的最小花费，g[u]=max(f[u][c])(c在u的子树内)。在递推f[u][c] 之前，我们先预处理出dis[i][j]，表示i,j两点在树上的距离。接下来分几种情况讨论(v表示u的儿子)。

当c不在u这棵子树内，

$$f[u][c] = sum(max(f[v][c], g[v])) + d[dis[u][c]]$$

当c在u的v0儿子的子树中，

$$f[u][c] = sum(max(f[v][c], g[v])) + f[v0][c] + d[dis[u][c]]$$

于是此题轻松解决。复杂度$O(n^2)$。

提示与补充：有兴趣的同学可以做做codechef Pruning Trees，方法是类似的。

4.8 156E Pancakes

4.8.1 Description

题意：有m种食谱，n种配料。配料从0 n-1进行编号，每种配料有一个权值ai。每种食谱的描述如下：di, si, ci，含义是在di进制下，si所表示的字符串由'?'和数字组成，'?'表示这一位可以任填。该食谱所需要的配料是那些编号可以和si匹配的配料，而该食谱的价格是v = ci+ π (ai)，现在请输出v的最小质因子，如果质因子¿100，则输出-1。

数据范围：$n \leq 10000, m \leq 30000, di \leq 16, len(si) \leq 30, ai, ci \leq 10^18$

难度：★★★★

4.8.2 Analysis

算法：数论优化

分析：我们把问题分为两步；第一步，求出v，第二步，求最小质因子。

第一步，实际上是很暴力的过程，按进制枚举所有食谱，然后求出它所有对应的配料，对食谱进行hash 判重即可。由于进行了判重，所以最后复杂度不会很高。

第二步，求最小质因子，一个显然的想法是将v mod 所有1 100的质因子的结果都算出来，这样复杂度可能要乘上100。。难以承受。我的优化方式是将所有的质因子分成了若干组，每组的乘积不超过10^9，先将每组的结果求出来，再分别在每组内尝试即可。最后我分成了四组，所以复杂度只需要乘上4。。可以通过全部数据。

提示与补充：质因子分组情况如下：

第一组：$2, 3, 5, 7, 11, 13, 17, 43, 97$

第二组：$29, 73, 79, 83, 89$

第三组：$47, 53, 59, 61, 67$

第四组：$19, 23, 31, 37, 41, 71$

4.9 105D Geodetics

4.9.1 Description

题意： 在此游戏中地图被分为了许多叫作Geo格的正方形方格，其中一些被涂上色，假设没有涂色的为透明色。

地图中还有些Geo符号，它们样子像不同颜色的金字塔（包括透明Geo符号）。每个Geo符号都坐落在Geo 格上，每个Geo格上最多一个Geo符号。

Geo符号可以被消除。为了更好地理解Geo符号在消除时发生了什么，这里引入把刚消除的Geo符号放入的队列。

从队列中取出Geo符号，观察包含Geo符号的Geo格的颜色，如果它不是透明的且颜色不同于Geo符号，则把所有这个颜色的Geo格重新涂为Geo符号的颜色（透明的Geo符号则为透明色）。重涂色是在一个无限大的区域从那个有符号的Geo格子开始螺旋状进行的。

25 10 11 12 13
24 09 02 03 14
23 08 01 04 15
22 07 06 05 16
21 20 19 18 17

换句话说，我们选择所有需要重涂色的方格找到它们在以有符号格为中心的无限螺旋表格中所对应的数字。此后按数字的增加顺序我们对其重染色。

如果在重染色时遇到一个格子包含另一个Geo符号的情况则将Geo符号移出并放置在队列尾部。

当重染色结束后Geo符号彻底消失，并且队列中下一个Geo符号（如果有）将取出，重复此操作直至队列为空。

你知道所有格子的颜色、所有符号的位置。计算出队列里符号彻底消失时所造成的重染色次数。

数据范围： $n \leq 300, m \leq 300$

难度： ★★

4.9.2 Analysis

算法： 并查集

分析： 本题的难点在于读题= 。。在理解好题意以后，题目就变得简单了。因为一个有geo符号的格子一旦被重染色后geo符号就会消失，所以有意义的染色只有n*m次，这个我们可以暴力。但是统计答案时实际上是统计某种颜色的格子有多少个，我们只需要用一个并查集维护每个颜色的集合的元素个数即可。复杂度O(n*m*log)。

提示与补充： 实现上会有一点复杂=。

4.10 226E Noble Knight's Path

4.10.1 Description

题意：给定一棵树，动态维护以下操作。

1、将一个顶点染成黑色。（每个顶点至多被污染一次，没有逆操作）

2、只考虑T时刻到现在的染色，寻找u到v的路径上第k个没有白色顶点。

数据范围：$n, m \le 10^5$
难度：★★★★

4.10.2 Analysis

算法：可持续化数据结构
分析：codeforces上少有的可持续化题目。我们首先考虑这只是一个序列的情况。也就是说在一个序列上进行染色，每次查询T时刻到现在时刻I的[l,r]的第k个白色点。

由于染色只会染一次，所以我们可以查询[1,I]中[l,r]被染色多少次-[1,T-1]中[l,r]被染色多少次= [T,I]中[l,r]被染色多少次。这个操作可以用可持续化线段树维护。通过一个类似于线段树上的二分操作，可以在logN的时间回答第k个白色点的位置。

如果转换成树结构呢？同样想办法将它转换成序列。括号序列！dfs时遇到一个节点时，将它加入栈，并且标记+1，当从一个点回溯时，也将它加入栈，标记-1。这样任意一个点都只出现了两次。如果查询u到它的某个祖先x的路径，只需要查询括号序列x的+1到u的+1即可。不存在于这条路径上的点将会在序列中出现两次，一次+1，一次-1。

通过这个办法，每次对u节点染色时，将+1处+1，-1处-1。维护时仍然维护黑点的个数即可。

当查询u到v的路径时，先查询u到lca中有没有第k个白色节点出现，再查询lca到v。

时间复杂度O(nlogn)。

提示与补充：关于可持续化的资料可以看看丽姐的论文。

Chapter 5

codeforces(5)

5.1 193D Two Segments

5.1.1 Description

题意： 有一个1到n的排列：a1,a2..an。现在从中选出互不相交的两段，使得这两段的并集排序后相邻两项相差为1，问有多少种选法。

数据范围： $n \leq 3 * 10^5$

难度： ★★★★

5.1.2 Analysis

算法： 数据结构

分析： 我们考虑这个问题的另一个等价模型，令b[a[i]] = i,原问题等价为，从b 数组选出连续的一段，使得这一段的元素能够被分成两个集合，每个集合满足排序后相邻两项相差为1。

我们维护s[i]代表b数组中从i开始的段当前最少能够被分为多少个集合，当$s[i] \leq 2$ 时，我们应计入答案。现在考虑往段的最后添加一个元素j，那么有一些s[i]会发生变化，为了考察变化的情况，定义$x1 = a[b[j] - 1], x2 = a[b[j] + 1]$，如果超出了数组范围，x 为0。

可以想象，x1和x2就是j在a数组中的邻居。令$x1 < x2$，可以发现在[x2+1,i]这一段，s[i] 会+1，在[x1+1,x2]这一段，s[i]不变，[1,x1]这一段,s[i]会-1。

所以现在的问题变成了：一个序列中，对某些段+1或者-1，询问权值$<= 2$的元素个数。

注意到整个序列的最小值不会<1，所以我们可以用线段树维护段的最小值，次小值及出现的次数。复杂度$O(nlogn)$。

提示与补充： 具体操作时可以将"最小值及出现次数"作为一个类进行操作。

5.2 76A Gift

5.2.1 Description

题意： 在一个n个点，m条边的无向图中。每个点有权值g,s，求该图的一棵生成树，使得生成树中所有边的$max(g) * A + max(s) * B$ 最小。

数据范围： $n \leq 200, m \leq 50000$

难度： ★★

5.2.2 Analysis

算法： 贪心

分析： 假设我们已经确定了max(g)的值，此时我们只需要将合法的边做最小生成树即可。

顺着这个思路思考，当max(g)稍稍增大时，我们将能够加入一些新的边，这些边对当前最小生成树有什么影响呢？假设我们加入的边是(u,v)，边权为s1，考虑下面两种情形：

1、u,v不在同一棵树中，这时只需要将(u,v)加入即可

2、u,v在同一棵树中，这时(u,v)的加入将会替换一条原本在u到v的链上的边，我们找到这条链上s值最大的边，判断是否能够替换。

维护好最小生成树后，更新一下当前答案即可。时间复杂度$O(n * m)$。

提示与补充： 无。

5.3 76F Tourist

5.3.1 Description

题意：在一个数轴上，有一名游客，他每秒能移动V的距离，他知道在ti时刻xi地点会发生事件，现在他想要经历尽可能多的事件。问：

1、0时刻从原点出发，能经历的最多事件
2、0时刻从任意点出发，能经历的最多事件

数据范围：事件数$\leq 10000, xi, ti \leq 10^8$
难度：★★★

5.3.2 Analysis

算法：DP
分析：我们令f[i]代表接到经历第i个事件时最多经历了多少个事件。很容易得到转移方程：

$$f[i] = max(f[j]) + 1(\text{当} ti - tj >= |xi - xj|)$$

对转移条件进行变形：

$$ti - tj >= xi - xj \quad and \quad ti - tj >= xj - xi$$

令xi = ti - xi, yi = ti + xi，得：

$$xi >= xj \quad and \quad yi >= yj$$

插入事件点(0,0)，我们将事件按xi排序进行DP，用树状数组即可维护好当前yi中的最大值。至此第一问已经完美解决。

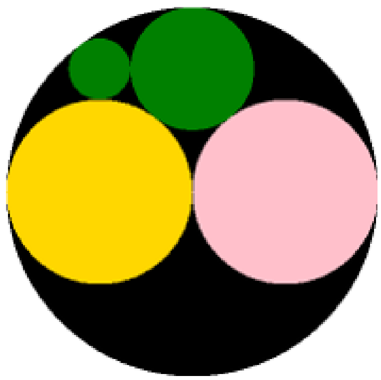
第二问可以从任意一个点出发，如何选择呢？倒过来做！定义f[i]为从i出发能经历的最多事件，从所有的f[i]中选择最大的一个即可。

提示与补充：2010年国家集训队互测中李其乐同学的“免费的馅饼”一题与本题类似。更详尽的题解可以参考李其乐同学的解题报告。

5.4 77E Martian Food

5.4.1 Description

题意：这是一个有趣的问题：在一个大小为R的圆O1中，放置一个大小为r的O2，使得O1和O2内切，再放置1个特殊圆O3，使得O3和O2外切，和O1内切，并且半径为R-r。接下来连续放置k个圆，第i个圆和第i-1个圆以及圆O2外切，与圆O1内切。默认第0个圆为圆O3。问第k个圆的半径。下图是k=1的例子：

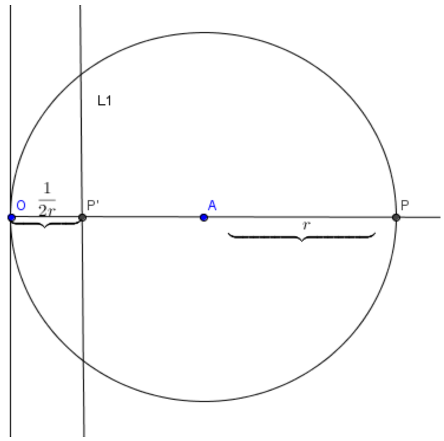


数据范围：T组数据，$T, k \leq 10^4$
难度：★★

5.4.2 Analysis

算法：反演
分析：几何反演的定义：在平面直角坐标系中，以原点O为中心，对于点P，它的反演变换之后的点为P'，满足$|OP| * |OP'| = 1$，且O,P,P'共线。

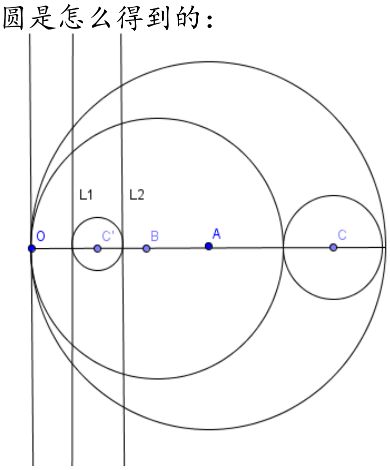
反演的性质：
1、过原点的圆反演之后为变为直线，如图所示（圆A反演为直线L1）：



(1)观察可以发现，在直线左侧的点由圆外的点反演得到，在直线右侧的点由圆内的点反演得到。

(2)如果有一个圆与该圆外切，则反演以后应与该直线相切（只有一个交点），并在该直线的左侧，如果内切则在右侧。

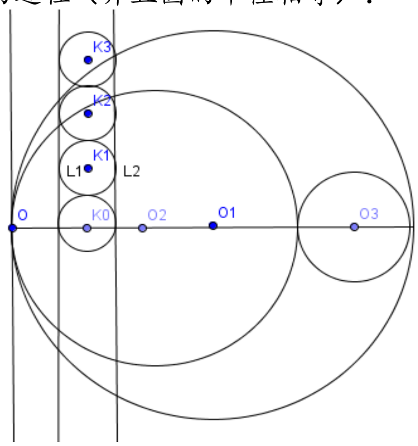
2、不过原点的圆反演之后仍然为圆，下图较为直观的告诉我们圆C的反演圆是怎么得到的：



找到与圆C内切的且过原点O的圆A，找到与圆C外切且过原点O的圆B，它们反演之后的直线分别为L1,L2，则圆C的反演圆在L1的右侧，在L2的左侧，且同时与这两条直线相切，也就是圆C’。

有了上面两组性质，再来解决这个问题相信并不困难。我们将上述图中的圆A和圆B改为题目中的圆O1，圆O2，那么圆O3的反演圆就是图中的

圆C'.新的圆的反演圆仍然要夹在两条直线中间，并且与上一次放入的圆相切（有两个方向，但是它们都等价），于是我们可以想象这是一个类似于堆圆的过程（并且圆的半径相等）：



于是第k个圆的圆心我们可以在O(1)的时间内得出，它将是$(\frac{R+r}{4*R*r}, \frac{k*(R-r)}{2*R*r})$。再将此圆逆反演回去，算出它的半径即可。

提示与补充：我也是第一次接触反演，上述一些描述是基于自己的体会，如果有不严谨的地方，希望赐教。另外反演在解决圆的相切问题时，确实较为方便，前提时你需要找到一个恰当的反演中心。

5.5 79D Password

5.5.1 Description

题意： 对于一个长度为n初始全部为0的序列，每次的操作为选定一个x($x \le L$)，再取连续的a[x]个格子进行取反。问最少多少次能够得到目标序列。保证目标序列1的个数<= 10。

数据范围： $n \le 10^4, L \le 100$

难度： ★★★

5.5.2 Analysis

算法： DP+贪心

分析： 本题曾出现在湖南2010年省队集训中。我们首先将该序列转化为它的差分序列，由于异或操作等价于mod2意义下的加法操作，所以我们每次操作[l,r]这一段，等价于改变两个点的状态：d[l], d[r+1]。

接下来考虑一种特殊的情形：目标序列的差分序列中只有两个1，分别在x1,x2位置。可以想象，这时候我们需要从x1出发，每次跳a[i]的距离，最后跳到x2,问最少所需要的步数。

从这种情形推广，我们可以得出，在一般情况下，我们只需要对所有的1进行两两配对即可。为了快速计算x1跳到x2的最少步数，我们需要预处理从每个点出发跳到其它任意点的最少步数。从每个为1的点出发bfs一次即可。复杂度$O(20 * n)$。

预处理完成后，我们还需要找到一个两两配对的最优方案。显然这是一个一般图的最大权匹配问题。由于点数很少，考虑进行压位。定义F[i][s]为前i个点，尚未配对的点的情况为s 的最优值，转移即可。复杂度大约为$O(20 * 2^{20})$。

提示与补充： 本题中差分的方法是很经典的解题手段。

5.6 81E Pairs

5.6.1 Description

题意：有n位同学，每问同学都有自己最喜欢的一位同学，现在要选出尽可能多的组，每个组包含两个同学，并且其中一个是另外一个最喜欢的，在此前提下，要求男女配对的组尽可能多(good idea)，问应该怎么选？

数据范围：$n \leq 10^5$

难度：★★

5.6.2 Analysis

算法：树形DP

分析：这是经典的环套树的DP。先考虑树的情形，定义f0[u]为以u为根的子树中u 尚未配对的最优值，f1[u]为u已经配对的最优值，最优值是一个二元组，第一维是总配对数，第二维是男女配对数。则

$$f0[u] = \sum max(f1[v], f0[v])$$

$$f1[u] = \sum max(f1[v], f0[v]) + f0[v0] - max(f1[v0], f0[v0])$$

对于环的处理，可以强制选择某条边，转化成树的情况再做。

提示与补充：无。

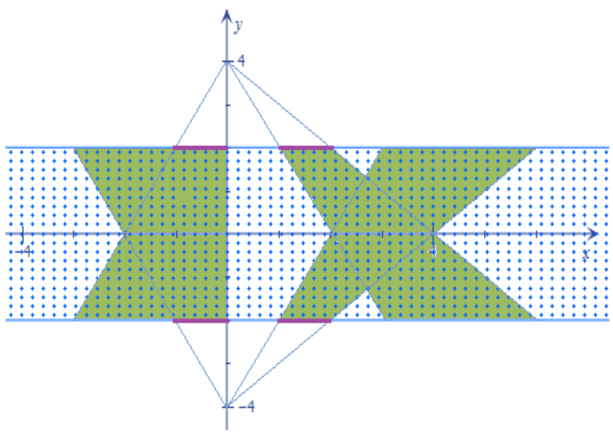
5.7 82E Corridor

5.7.1 Description

题意： 下图是一个房子的水平面。

我们用$-h \le y \le h$代表的区域来表示这个房子的水平面。在有两个点光源$(0, f), (0, -f)$，光源严格处于房子外。房子的墙上有窗户，我们用$y = h$和$y = -h$上的线段表示窗户，且窗户是关于x轴对称的。

求房子里被照亮的区域的面积。



数据范围： $n \le 500$

难度： ★★

5.7.2 Analysis

算法： 计算几何

分析： 定义由上往下照所形成的梯形叫上梯形，由下往上照所形成的梯形叫下梯形。首先可知上梯形，下梯形自身不会相交。所以我们可以采取这样一种方法：

1、统计所有上梯形的面积

2、对于每一个下梯形，求出所有上梯形交它的补集。

求和即可。求两个梯形的交的面积可以分别用半平面去交梯形再取补集（类似于半平面交的暴力做法）。

提示与补充： 实现上稍有复杂。

5.8 83E Subsequences

5.8.1 Description

题意： 定义s1,s2为0,1串，定义函数$f(s1, s2) = min(len(s3))$,s3满足前缀匹配s1,后缀匹配s2。f(s1,s2,s3) = f(f(s1,s2),s3)，依次类推。现在给定n个0,1串，请将它分成两个子序列，使得两个子序列的f值相加最小。

数据范围： $n \leq 2 * 10^5, len(s) \leq 20$

难度： ★★★

5.8.2 Analysis

算法： DP

分析：

1. 定义f[i]为前i个串，第i个和第i-1个串属于不同子序列时f值相加最小的值。
2. sum[i]为前i个串属于同一个子序列时f值为多少。
3. get(i,j) 表示求f(s[i], s[j])函数的值。

则我们可以得到递推方程：

$$f[i] = min(f[j] + sum[i-1] - sum[j] + get(j-1, i))$$

但是这个方程是$O(n^2)$的复杂度，注意到get(j-1,i)只有20个取值，所以可以想到优化方法定义g[k][s]为当前有s后缀的f[j]-sum[j]最大为多少。每次查询只需log(n),维护也只需要log(n)。

提示与补充： 无

5.9 85E Guard Towers

5.9.1 Description

题意：平面上有n个点，现在需要将点分成两个集合，使得两个集合内曼哈顿距离相差最大的点距离尽可能小。问距离和方案数。

数据范围：$n \leq 5000, 0 \leq xi, yi \leq 5000$

难度：★★

5.9.2 Analysis

算法：并查集or 贪心

分析：

方法1：对n个点两两求距离，对距离进行从大到小的排序（可以用桶排）。接下来用并查集检查能否分成两个集合，一旦不能分成两个集合，就得到了答案，而方案数是当前并查集中$2^{联通块的个数}$。

方法2：一种十分巧妙的方法。我们令$x0 = x + y, y0 = x - y$，可以发现原问题的曼哈顿距离$|x1-x2|+|y1-y2|$等价于转换后的$max(|x1-x2|, |y1-y2|)$，也就是说我们将把点集分成两个矩形，答案是矩形边长的最大值。可以发现，两个矩形一定分别覆盖了整个点集的大矩形的两条边，所以我们枚举覆盖的情况，即可求得距离的答案。至于方案数，我们只需要再枚举一遍点，看能够放到哪个矩形即可。

提示与补充：注意两种覆盖情况得到的距离相等的情况。

Chapter 6

codeforces(6)

6.1 86E Long Sequence

6.1.1 Description

题意： 给定c[1],c[2]...c[k]，定义a序列满足

$$a[n] = (c[1] * a[n-1] + c[2] * a[n-2]... + c[k] * a[n-k])mod2$$

当确定a[0]..a[k-1]时，整个a序列就确定了。并且a序列一定是一个循环序列。现在给定c 序列，请你构造一个a序列的前k项（不全为0），满足它的最小正周期为$2^k - 1$。

数据范围： $k \leq 50$
难度： ★★

6.1.2 Analysis

算法： 矩阵乘法
分析： 先思考这样一个问题，当a序列的前k项确定，如何判断它的最小正周期呢？

首先，我们需要计算一个常系数递推序列的第n项，这是一个非常经典的问题，可以用矩阵乘法在$O(k^3 * logn)$的时间内实现。

接下来，我们检验$2^k - 1$是否是a序列的周期，求出从2^k开始连续k项即可。

然后，我们再判断$\frac{2^k-1}{pi}$是不是a序列的周期(pi为2^k-1的质因子)，这样便可得知2^k-1是不是最小正周期。

于是我们可以在$O(k^4 * log)$的时间内检验一个a序列，那么如何构造呢？。。随机即可。因为据经验这样成功的概率很高= 。。。

提示与补充： 无。

6.2 89D Space Mines

6.2.1 Description

题意： 三维空间中分布着n个球体，每个球体从球心发出不超过m条线段，线段的长度d满足$r \leq d \leq 1.5r$。现在一个特殊的球（死星）从某点沿某方向匀速前进，问它是否会碰到n个球体，或者球体发出的线段，如果碰到，第一次碰到的时间是多少？

数据范围： $n \leq 100, m \leq 10$

难度： ★★

6.2.2 Analysis

算法： 计算几何

分析： 我们对每个球体一一求出相撞时间，再取最小值即可。

对于球体，我们判断两个球心是否会存在距离小于半径之和的情况。

对于线段，我们只需要判断线段端点有没有可能撞到死星。这是因为当死星撞到线段的非端点时，它一定已经和该球相撞（可以通过不等式解出）。

提示与补充： 无。

6.3 91D Grocer's Problem

6.3.1 Description

题意： 现在有一个排列a1,a2...an，你每次的操作为：选出不超过5个位置，重新安排它们的值（可以打乱顺序）。你的目标是让整个序列变为1,2,3...n，问至少需要几次操作？

数据范围： $n \leq 10^5$

难度：★★

6.3.2 Analysis

算法：贪心

分析：我们首先将本题转换为图论模型，如果ai=j，则令i向j连一条边。这样，整个图变成了若干个大小不一的环。

我们用如下的一个算法流程进行贪心：

1. 当一个环的长度>= 4,我们取出该环中连续的5个，令4个归位，也就是环的长度-4。如果没有5 个，则直接归位。
2. 接下来只剩下3元环和2元环，如果同时存在2元环和3元环，则同时消去，直到不存在为止。
3. 如果有两个3元环，将一个3元环消去，同时将另一个3元环变为2元环，跳回步骤2。
4. 将剩下的2元环两两消去。
5. 将剩下的3元环消去。

可以证明，上面的贪心是正确的，因为每次都尽可能多的减少了环的长度。

提示与补充：无

6.4 93D Flag

6.4.1 Description

题意：有一个长度为n的序列，每个格子的颜色为黄色，黑色或者白色，并且满足如下要求：

1. 相邻的格子颜色不能相同。
2. 白色和黄色的格子不能相邻。
3. 红色和黑色的格子不能相邻。
4. 不能存在3个连续的格子为：黑白红或者红白黑。
5. 两个对称的序列被认为是等价的。

你的任务是求出长度在L到R之间的不等价合法序列有多少个。输出$mod(1e9+7)$的值。

数据范围：$L, R \leq 10^9$
难度：★★

6.4.2 Analysis

算法：矩阵乘法
分析：用f[i][s1][s2]代表前i个格子最后两个格子的颜色分别为s1,s2时，合法序列的数量。

如果不考虑对称等价的条件，可以轻松建出转移矩阵，由f[i]转移到f[i+1]。再通过矩阵乘法求得f[n] 的答案。

考虑对称的情况，计算$\frac{f[n]+f[(n+1)/2]}{2}$即为去掉等价情况以后的合法序列个数(加上回文串后每个串都被计算了两次)。

套上前缀和转移，可以快速求出长度$<=R$的合法序列的个数，再减去$<=L-1$的序列即可。

复杂度$O(3^3*logn)$。

提示与补充：无

6.5 97C Winning Strategy

6.5.1 Description

题意： 定义a_k序列满足:$a_k \le \sum_{i=1}^{k-1}(n-2a_i)$，且$0 \le a_k \le n$。
现在给定pi序列p0,p1..pn，满足$0 <= p0 <= pi <= pi+1.. <= pn <= 1$，
请你安排一个无穷数列ai, 使得下式最大：

$$\lim_{m->oo} \frac{\sum_{i=1}^{m} p_{ai}}{m}$$

数据范围： $n \le 100$
难度： ★★

6.5.2 Analysis

算法： 贪心
分析： 这个式子看起来实在颇为复杂。我们不妨用下面的方式理解它：

1、当我们选择的a_k满足$a_k <= \frac{n}{2}$时，它会为未来提供$n-2*ai$的空余量。
2、当我们选择的a_k满足$a_k > \frac{n}{2}$时，它将会占有$2*ai-n$的空余量。

可以想象，我们每次可以提供或者占有一定的空余量，并且对答案贡献一定的概率。我们应该选择最有效的那个！。

所以可以找到提供空余量中最有效的与占有空余量最有效的a_k进行组合，即是最后答案。
提示与补充： =。

6.6 97A Domino

6.6.1 Description

题意： 小Gennady生日时收到了一副骨牌。这副骨牌包含28块2*1的不同的骨牌，骨牌的每一半包含一个0到6的数字。如下所示：

0-0 0-1 0-2 0-3 0-4 0-5 0-6
1-1 1-2 1-3 1-4 1-5 1-6
2-2 2-3 2-4 2-5 2-6
3-3 3-4 3-5 3-6
4-4 4-5 4-6
5-5 5-6
6-6

一个包含了28个骨牌的图被叫成magic，仅当能被14个互不相交的2*2的方块覆盖，每个方块能由4个相同的数字组成。

Gennady选择了一个n*m的棋盘，并在上面放了许多1*2的矩形小片。每个小片恰好占领了棋盘上的两个相邻方格，这些小片不能重叠，但可以互相接触。棋盘上总共有28个小片，和骨牌个数相同。现在，Gennady要把每个小片换成骨牌，拼出magic图片。不同的小片应被替换成不同骨牌。请算出在小片位置给定的情况下，Gennady能有多少种合法的替换方案。

数据范围： $n, m \le 30$
难度： ★★

6.6.2 Analysis

算法： 搜索
分析：

如上图所示，当我们确定好最后颜色相同的矩形是怎样一种配对情况后，标号其实是可以互相替换的。也就是说我们关心的实际上只有配对情况。

于是可以将配对情况搜索出来并且判断是否合法，如果合法对答案的贡献为7!（因为替换等价）。

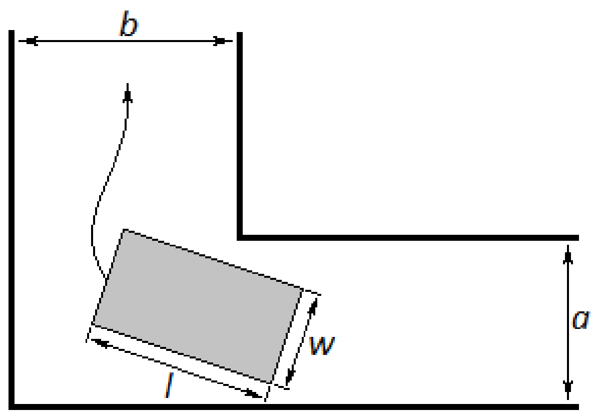
复杂度为O(13*11*9*7*5*3*O(check))。

提示与补充： 无

6.7 98C Help Greg

6.7.1 Description

题意：



如题所示，我们要将一个长方形拖过这个拐角。给定拐角的边长分别为a, b，长方形的长为l，问长方形的宽最大为多少？

数据范围： $a, b, l \leq 10^4$

难度： ★★

6.7.2 Analysis

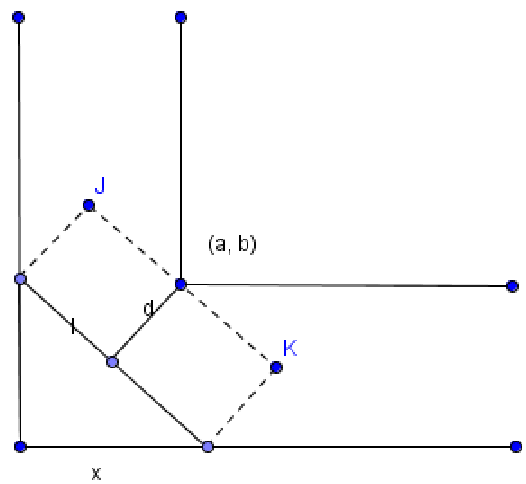
算法： 三分

分析： 分情况讨论(假定$b < a$)：

1、当$l < b$时，宽最大为l

2、当$b <= l < a$时，宽最大为b

3、当$b < a < l$时，我们建立平面直角坐标系：



可以想象，我们将该矩形拖过去时，在两个角都抵死的时候会最优。所以当x确定时，我们令d=f(x)，目标则是要求f(x)的最小值，因为一旦宽度大于f(x)，就无法通过了。f(x)是一个单峰函数，可以三分求得最小值。

提示与补充：无

6.8 98D Help Monks

6.8.1 Description

题意： 汉诺塔问题的加强：N个盘子，在第一根柱子从小到大的叠着，可能有的盘子大小相同。要求最后移动到第三根柱子，在移动过程中，不允许出现大的盘子在小的盘子上方，要求最后移动到第三根柱子时，顺序和第一根柱子一样。问最少移动步数，以及一种方案。

数据范围： $n \leq 20$

难度： ★★★

6.8.2 Analysis

算法： DP

分析： 定义f[n][A][B][C]为前n个盘子从A柱转移到B柱，顺序完全一样的最少步数。g[n][A][B][C]为转移后，最后一段大小相等的盘子完全倒序的最小步数。

可以按照如下方式转移，设最后连续的大小相等的盘子有m个:

1、考虑求g[n][A][B][C]，将前n-m个盘子倒序移到C柱，再将最后m个倒着移动到B柱，再将n-m个盘子移回C柱。此时最后m个是倒着的。

$$g[n][A][B][C] = g[n-m][A][C][B] + m + g[n-m][C][B][A]$$

2、考虑求f[n][A][B][C]，有两种情况：

(1)最后一次移动1个:

$$f[n][A][B][C] = g[n-1][A][C][B] + 1 + g[n-1][C][B][A]$$

(2)最后一次移动m个到C，再将m个从C移动到B:

$$f[n] = g[n-m]*2 + f[n-m] + m*2$$

可以用归纳法证明上述转移的正确性。

提示与补充： 无

6.9 191D Metro Scheme

6.9.1 Description

题意：在一个n个点m条边的点仙人掌图中，用路径和环进行覆盖。问最少和最多的使用量分别是多少？

数据范围：$n \leq 10^5, m \leq 3 * 10^5$

难度：★★

6.9.2 Analysis

算法：贪心

分析：最多的使用量输出边数即可，我们只考虑最少的使用量。

1、如果图中没有环的情况，我们统计出所有的奇度点，再两两配对，一定能够把所有的路径都覆盖掉。

2、如果一个环中有两个点的度数> 2，则一定能在第一种情况中解决（考虑到达环的路径延伸过去把环覆盖掉）。

3、如果一个环中只有一个点或者没有点的度数> 2，则必须再加一条路径进行覆盖。

按上面的规则进行贪心即可，复杂度O(n+m)。

提示与补充：无

6.10 164D Minimum Diameter

6.10.1 Description

题意： 给定平面上的n个点。你需要去删掉恰好k个点($k < n$)，使得剩下的n-k个点所构成的集合的直径尽量小。一个点集的直径是指集合中最远点对的距离。只含一个点的集合的直径等于0。输出删点的方案。

数据范围： $n \leq 1000, 1 \leq k \leq 30, k < n$

难度： ★★

6.10.2 Analysis

算法： 二分答案+搜索

分析： 我们将所有点两两计算距离之后排序。然后二分答案。如果某条边(x,y)的距离大于该答案，则(x,y)之间建一条边。

于是问题等价成选择一个图中尽可能少的点，使得每条边至少有一个点被选。（一般图的最小点覆盖问题）

这是一个NP问题，只能采用搜索的方式。有如下的几个剪枝可供参考：

1、在整个图中如果有一个度数为1的点，则我们可以一定不选它，而选择与它相连的那个点（一定不会变差）。

2、当前最大度数乘剩余可选择点数<当前边数，则不合法。

3、每次可以选择度数最大的点搜索。

最后能够在较快时间内通过全部数据。

提示与补充： 无

Chapter 7

codeforces(7)

7.1 150E Freezing

7.1.1 Description

题意： 在一棵带边权n个点的树中，要求选一条长度在$[l, r]$之间的路径，使得该路径上所有边的边权的中位数最大。

数据范围： $n \leq 10^5$

难度： ★★★

7.1.2 Analysis

算法： 树的点分治

分析： 二分答案以后，设答案为k，我们将树上$> k$的边权设为1，$< k$的边权设为-1。现在的问题就变成了在树上找一条长度在[l,r]之间路径，使得路径边权和$>= 0$。

对于这个问题，我们可以进行点分治。每次找到树的重心（重心的定义是：最大子树最小），将树剖分成若干棵子树，再分别递归下去，因为每次子树的大小至少减少一倍，所以递归层数只有$logN$层。

如何合并呢？如何统计过根节点的路径中有没有边权和$>= 0$的？可以枚举一棵子树，再在另外一棵子树利用单调队列找出合法区间的最大值。

算法复杂度O(nlogn)。

提示与补充： 可以在点的分治中进行二分，这样常数会小很多。

7.2 101E Candies

7.2.1 Description

题意： 定义函数

$$f(a,b) = (x_a + y_b) \quad mod \quad p + max(f(a,b-1), f(a-1,b))$$

$$f(0,0) = 0$$

给定a,b，求f(a,b)的值及方案。

数据范围： $a, b \leq 20000$，时限15s，内存45MB
难度： ★★

7.2.2 Analysis

算法： DP
分析： 如果只需要求出f(a,b)的值，本题便变得非常简单，只需要利用滚动数组根据定义进行转移即可。

难点在于需要求方案，更直接的说，难点在于不能够开一个20000*20000的数组。

解决方案？分块！我们将a看成行，b看成列。求方案时，我们需要从(a,b)逆推回来。可以每隔100行，存下该行的f函数，逆推回来时，利用本行的函数值，再往上推100行即可，这样仅需要100*20000的空间，足以通过此题。

提示与补充： 无

7.3 103E Buying Sets

7.3.1 Description

题意：给定n个集合, 要求选出其中某些集合, 使得这些集合的并集的势, 等于选出的集合的数目.

对于任意的k($1 <= k <= n$), 满足从中选出任意k个集合, 这k个集合的并集的势一定大于等于k.每个集合有一个权值, 每个选择方案的代价是所选的集合的权值的和.

请输出代价最小的选择方案的代价.

数据范围：$n \leq 300$, 权值大小$\leq 10^6$
难度：★★★

7.3.2 Analysis

算法：最小割
分析：由于任选k个集合的并集的势都大于等于k，所以我们的目标变为任选k个集合，使得它们的并集的势尽可能小，在此前提下权值和尽可能小。

对该问题进行一下变形：当我们每选定一个集合时，可以得到V的权值(V>所有元素的权值之和)，而每一个元素被选中时，将会得到-（V的权值+ 它本身的权值），现在求一种方案，使得选出的权值尽可能大。

当某个集合选中时，集合里的元素也将被选中。这是一个典型的最大权封闭子图问题。我们能够借用网络流模型来解决它。

新建点S,T，由S向所有正权点连边，边的容量为权值，所有负权点向T连边，边的容量为-负权。点与点之间的边保留不变。从S到T求一遍最大流流。用$\sum$正权-maxflow即为答案。关于该算法的正确性可以参见2007年胡伯涛同学的集训队论文。

提示与补充：不选任意集合也是一种方案。

7.4 105E Lift and Throw

7.4.1 Description

题意：给定一条标有整点(1, 2, 3, ...)的射线. 定义两个点之间的距离为其下标之差的绝对值.

Laharl, Etna, Flonne一开始在这条射线上不同的三个点, 他们希望其中某个人能够到达下标最大的点.

每个角色只能进行下面的3种操作, 且每种操作不能每人不能进行超过一次.

1) 移动一定的距离

2) 把另一个角色高举过头

3) 将举在头上的角色扔出一段距离

每个角色有一个movement range参数, 他们只能移动到没有人的位置, 并且起点和终点的距离不超过movement range.如果角色A和另一个角色B距离为1, 并且角色B没有被别的角色举起, 那么A就能举起B. 同时, B会移动到A的位置,B原来所占的位置变为没有人的位置. 被举起的角色不能进行任何操作, 举起别人的角色不能移动.

同时, 每个角色还有一个throwing range参数, 即他能把举起的角色扔出的最远的距离. 注意, 一个角色只能被扔到没有别的角色占据的位置.我们认为一个角色举起另一个同样举起一个角色的角色是允许的. 这种情况下会出现3个人在同一个位置的情况.根据前面的描述, 这种情况下上面的两个角色不能进行任何操作, 而最下面的角色可以同时扔出上面的两个角色.

你的任务是计算这些角色能够到达的位置的最大下标, 即最大的数字x, 使得存在一个角色能够到达x.

数据范围：所有输入在[1,10]之间的整数。
难度：★★★

7.4.2 Analysis

算法：记忆化搜索
分析：题目真难读= 。一般题目比较难读的话，算法就会比较简单。。因为cf的比赛只有那么长的时间嘛。我们首先不妨计算一下该题的状态数最

多会有多少。

每个角色所处的下标范围最多在[1,60]之间（应该达不到这个范围），每个角色本身有三个技能，在同一位置时，还需要记录角色的上下位置关系。并且这个状态是远远达不到的！想到这里，我们可以大胆的开始记忆化搜索了。。

对状态进行hash，以使它能存下来=。然后剩下来的就是颇为复杂的转移了。只要你有足够的耐心，本题对你已经没有难度。

提示与补充： 无

7.5 107D Crime

7.5.1 Description

题意： 一个长度为N的字符串（仅由大写字母构成）。有m个条件，每个条件为c,d，表示c这个字符出现的次数应是d的倍数。如果两个条件的c相同，那么c只需要满足其中某一个条件即可。问满足所有条件的字符串有多少个？输出它mod12345的值。
数据范围： $n \leq 10^{18}, m \leq 1000, \prod d \leq 123$
难度： ★★★

7.5.2 Analysis

算法： 矩阵乘法
分析： 注意一个很关键的条件：$\prod d \leq 123$。这是我们能够使用矩阵乘法的原因。

定义pi为第i个字符出现的所有条件中d的乘积，定义f[i][c1][c2][ck],为长度为i的字符串，A字符出现的次数$\bmod p1 = c1$，B字符出现的次数$\text{mod} p2 = c2...$的方案数。

由f[i]转移f[i+1]是很容易的，并且转移矩阵的边长不会超过123。

所以我们可以在$O(logn * 123^3)$内求出f[n]的矩阵。注意对于同一个字符，如果有多个条件，我们只需要满足一个即可，所以我们再检验一下每个状态是否满足某一个条件，如果满足即为合法，我们便把它贡献到答案。

7.6 115D Expression

7.6.1 Description

题意： 我们定义UAE (unambiguous arithmetic expression) 为

1. 所有的自然数是UAE,有前导零的自然数(比如0000,0010)也是UAE
2. 如果X和Y是UAE, 那么”(X)+(Y)”,”(X)-(Y)”,”(X)*(Y)”,”(X)/(Y)”也是UAE
3. 如果X 是UAE, 那么”-(X)”和”+(X)”是UAE

现在给你一个只包含’0’-‘9’和’+’,’-’,’*’,’/’的字符串，让你计算有多少种不同的UAE,满足去掉了括号符号后就变成了输入中的串，答案Mod 1000003输出。

数据范围： $n \leq 2000$
难度： ★★★

7.6.2 Analysis

算法： 组合数学
分析： 一看到此题，首先想到的是$O(n^3)$算法。定义f[l,r]代表l到r的方案数，再枚举最后运算的符号在第k位，则

$$f[l][r] = \sum f[l][k-1] * f[k+1][r]$$

这样的一个式子似乎没有什么优化空间，我们得另辟蹊径。

思考一下，本题的瓶颈在于出现了连续的符号，如果没有这种情况，我们大可以将状态变成一维f[n]，并且最后的答案就是卡特兰数列的第n 项。

尝试将此题向卡特兰数列进行靠近，对它做如下的一个“去括号”变换：找到[l,r]最后运算的符号k，保留l匹配k-1的括号，去掉k+1匹配r的括号。这样一种变换和原问题是一一等价的，并且剩下的括号数为n-1个。每一个括号对应着一个符号。更确切的说，第i个右括号便对应着第i个符号。

变换以后实际上得到是一个括号序列，那么原问题中“连续的符号”在变换后有一个怎么的限制呢？如果第i个符号是一个”连续的符号”，那么第i个右括号前面必定是一个左括号！这是因为如果是连续的符号，那么便意味着它左边的表示式为空。

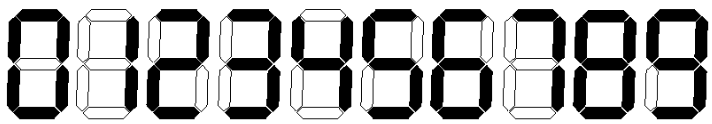
经过这样一番辛苦的转换后，我们终于得到了一个$O(n^2)$的算法，模拟括号序列，每次添加一个左括号或者添加一个右括号，保证“连续的符号”所对应的右括号前必定是左括号，求括号序列的方案数。

提示与补充：代码很短，这样一种神奇的转换，正反应了组合数学中深不可测的等价变形 。

7.7 120I Numbers

7.7.1 Description

题意： Vasya定义了一种火车票的幸运度定义方式如下：
每张票据由2*n位数字组成,所有数字由以下方式表示



这些如同电子钟上的一样的数字由七段线段组成,有的有颜色,有的没颜色.有颜色的表示了一个数字.Vasya把右边的n个数字迭在左边上,也就是第n+1个数字迭在第1个上,第n+2个数字迭在第2个上⋯..

对于每一对迭在一起的数字, Vasya统计在两个数字中都有颜色的线段的条数,把n个条数加起来,得到幸运值.

现在给你一个2*n位的数字,请你求出最小的一个大于它的2*n位的数字,使其幸运值也大于它的幸运值.

数据范围： $n \leq 10^5$
难度： ★★★

7.7.2 Analysis

算法： 数位统计
分析： 对于此类数位统计问题，我们只需要能够快速的做下面这件事情就可以了：
确定了前i位的数字，后2*n-i位任选，幸运值最大为多少？
1、当$i > n$时，我们找到[i-n,n]位对应的最优数字，相加即可得到最大幸运值。

2、当$i <= n$时，我们需要[1,i]位对应的最优数字，以及[i+1,n]位能够得到的最优匹配（即7），相加得到最大幸运值。

能够求出快速求出这样一个函数，我们就只需要从后往前一位一位尝试到可行解，再从前往后确定最优解即可。

提示与补充： 无

7.8 123E Maze

7.8.1 Description

题意：一个含有n个点的迷宫是一棵树（一个任意两点之间都恰好有一条路径的无向图）。每个点都有一定的概率成为这个迷宫的入口和出口。

从这个迷宫走出去的方法是从入口开始进行深度优先搜索。如果当前有多个移动方案，那么等概率的选择移动方案中的一个。DFS的过程为以下的伪代码：

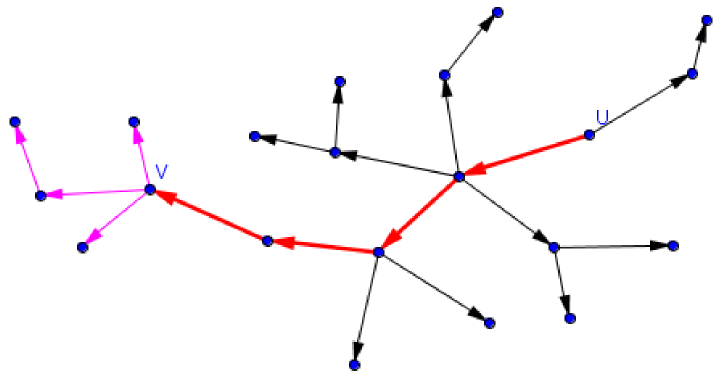
DFS(x)
if x == exit vertex then
finish search
flag[x] <- TRUE
random shuffle the vertices' order in V(x) // here all permutations have equal probability to be chosen
for i ¡- 1 to length[V] do
if flag[V[i]] = FALSE then
count++;
DFS(y);
count++;

V(x)是与x点相邻的点的序列。Flag数组初始时是全部为FALSE的。DFS初始时从入口开始。当搜索结束时，变量count将会统计移动的次数。你的任务是统计一个人从这个迷宫的入口走到出口步数的数学期望值。

数据范围：$n \leq 100000$
难度：★★

7.8.2 Analysis

算法：概率论
分析：考虑现在u点为入口，v点为出口的情形：



我们尝试统计每条边被遍历的概率。对于红色的边，一定会被计算1次，对于粉红色的边，一定不会被计算，而对于黑色的边遍历的概率是多少呢？当我们从路径上拐进了该黑色边所在的子树时，则该边被计算两次，否则不计算。拐进去的概率实际上就是0.5，因为先走路径上的边，还是先拐进去，实际上是等价的问题。

于是：除v点的后继边之外的边都对答案贡献了1。所以我们只需要枚举v点，然后计算答案即可。

提示与补充： 无

7.9 125E MST Company

7.9.1 Description

题意：在Berland有n个城市，m条道路，一些城市之间有道路连接。每个道路都有一个权值，而且没有方向限制。MST小组应该对某些道路执行修复工作，使得任意两个城市能只通过修复了的道路就能联通。此外，被修复的道路里应该包括K条与1连接的路。请你找出使得满足条件的权值和最小的修复道路集合。

数据范围：$n \leq 5000, m \leq 10^5$
难度：★★

7.9.2 Analysis

算法：图论
分析：经典的单点k度限制最小生成树。

一般的做法是这样的：

1. 将与1相连的边删去
2. 求出剩下图的最小生成树
3. 对于每条与1相连的边，如果将它加入当前最小生成树，则会替换一条边，设这样操作的增量为v
4. 选出增量最小的边加入，然后维护其它边的增量
5. 重复k次即可

每次都是最优的策略，复杂度为$O(n^2 + mlogm)$。

对于本题还有一个基于二分的做法，每次在与1相连的边上加一个增量，求最小生成树，看方案中与1相连的边的边数决定接下来增量应该增加还是减少。对于这个方法有一个小问题。可能不会刚好二分到k的解。也就说当增量为x时，边数比k小，增量为x+1时，边数比k 大。此时该问题需要做一些较繁琐的处理。

提示与补充：无

Chapter 8

codeforces(8)

8.1 193E Fibonacci Number

8.1.1 Description

题意： Fibonacci数列模10^13下是这样定义的：

1、F0 = 0, F1 = 1

2、$Fi = (Fi - 1 + Fi - 2) \quad mod \quad (10^{13})$ （当$i > 1$）

John想知道x是否在这个数列当中出现过，如果出现过，最早出现在哪个位置。

数据范围： 保证答案在2^{63}以内
难度： ★★★

8.1.2 Analysis

算法： 数论
分析：
定理1：Fibonacci≡ m意义下的周期定义为t(m),则t(a*b) = t(a) * t(b)（如果(a,b)=1）

说明：根据中国剩余定理可以证明。

定理2：$t(10^i) = 1.5 * 10^i$，（当$i >= 3$）

说明：上式可以通过打表发现规律。

于是我们首先暴力求出10^3下x出现的位置。接着通过这些位置扩展到10^4，保留下仍然有效的位置。重复上述过程直到10^{13}。

每次求第n个位置的数用矩阵乘法，由于同一个数字不会出现太多的次数，所以可以通过全部数据。

提示与补充：关于t(m)函数还有更多优美的性质，甚至可以在很快的时间内求出任意m的t(m) 函数，具体做法参考wiki。

8.2 145D Lucky Pair

8.2.1 Description

题意： 幸运数字是那些仅由4和7构成的数字。

现在有一个长度为N的非负整数序列A，其中A[i]表示该序列的第i个元素。要整个序列中选出两个互不相交的子段A[L1,R1], A[L2,R2]($1 <= L1 <= R1 < L2 <= R2 <= n$)，使得不存在某个幸运数字既在A[L1,R1]出现，又在A[L2,R2]出现。求有多少种选择方案。

保证A序列中幸运数字的总出现次数不会超过1000次。

数据范围： $n \leq 10^5$
难度： ★★★

8.2.2 Analysis

算法： 单调队列
分析： 可以发现，本题的核心在于幸运数字，所以我们可以先假设数据规模在1000的范围，并且只有幸运数字出现。

考虑这样一种情形，当我们固定了r1的位置，将l1不断向左推进时，在r1的右侧会出现越来越多的"断点",这些断点是指已经在[l1,r1]当中出现过了的数字。而[l2,r2]就必须在没有跨越断点的段中选择。

如果我们用set维护r1右侧的分段情况，就能够在$O(n^2logn)$的时间内解决这个问题。

接下来我们考虑一种更为巧妙的$O(n^2)$做法。将r1左侧的幸运数字在右侧进行标号，如果在左侧是l1 位置，则在右侧出现同样数字的地方标上r1-l1。这意味着当我们确定要选择[l2,r2]时，l1能够扩展的最长长度应是[l2,r2]中标号最小的那一个。所以我们采取这样一种单调队列的做法，在确定r1后，从右往左枚举l1，并且维护好关于标号的从大到小的单调队列，以及[l1,n]的标号最小值的和。这样就能够在$O(n^2)$的时间将此题完美解决了。

由于原题的数据规模是10^5，所以我们需要先将没有用的数字压缩成一个。
提示与补充： 无

8.3 132E England

8.3.1 Description

题意： 你被要求输出一个由n个数组成的序列。有m个变量给你用，变量必须是单个小写英文字母。

你能做的操作只有2种：

1. 对某个变量赋值
2. 输出某个变量

对于某个你给出的操作序列，定义其代价如下：每个赋值操作的代价是你想要赋给这个变量的值的二进制表示中1的个数。每个输出操作的代价是0。

你被要求给出一个操作序列以输出这n个数。同时，必须最小化操作序列的代价。输出这个最小代价，以及能达到最小代价的满足条件的操作序列。

数据范围： $n \leq 250, m \leq 26$

难度： ★★

8.3.2 Analysis

算法： 费用流

分析： 为了最小化代价，一个变量应该尽可能被使用多次。考虑建立如下的网络流模型：

n个数拆点变成2*n个点，每个数有一个入点和出点。入点向出点连边，费用为-oo，容量为1。对于两个数$x, y[x < y]$，由x的出点向y的入点建一条边，如果a[x]等于a[y]，该边费用为0，容量为1。如果不相等，则费用为a[y]二进制中1的个数，容量为1。

在新建S,S1,T，由S向S1连费用为0，容量为m的边，S1向所有点连边，容量为1，费用为0，所有点向T连边，费用为0，容量为1。

我们求该图S到T的最小费用最大流即可。最后的答案为cost+n*oo。

正确性说明：我们用不超过m条路径覆盖了n个点，使得总花费最小。

提示与补充： 无

8.4 138D World of Darkraft

8.4.1 Description

题意：给定N*M网格图，每个格子都是L,R,X其中某个元素。每个格子都要么是活跃的，要么不是活跃的，初始所有格子都是活跃的。

有两人玩回合制游戏，每回合当前操作的人选择一个活跃的格子：

- 如果格子上是L，格子以自身为起点，向左下方到右上方对角线方向发射2条射线。
- 如果是R，格子以自身为起点，向右下方到左上方对角线方向发射2条射线。
- 如果是X，格子将发射上述4条射线。

射线经过的格子将变成非活跃的，而射线遇到非活跃格子或者边界时会停止。轮到谁无法选择一个活跃的格子时那个人就输了。请判断先手是否必胜。

数据范围：$n, m \leq 20$
难度：★★★

8.4.2 Analysis

算法：博弈论
分析：这是一个典型的组合游戏，可以用sg函数求解。

观察可以发现，我们对网格图黑白染色后，黑色与白色之间是互不干扰的。所以我们将图分开来做。

将整个图旋转后可以发现：每次选择，要么将图分成四个部分，要么将图分成两个部分，也就是说游戏被分成了若干个子游戏。我们可以记忆化存下在[l1,l2,r1,r2] 边界下的sg 函数值，利用sg的定义进行推导即可。

状态数为$O(20^4)$，转移为$O(20^2)$，并且达不到这个级别。

提示与补充：关于sg函数的研究可以参见2009年贾志豪同学的《组合游戏略述——浅谈SG游戏的若干拓展及变形》。

8.5 140F Snowflake

8.5.1 Description

题意：定义：一个点集是中心对称的，指存在一个点X（不一定要属于这个点集），对于任意点集中一点a，一定有点集中某点b（b可以和a相同），使得b是a关于X的对称点。此时称X为点集的对称中心。现给定平面内N个点的集合，你可以添加最多K个点（也可以不添加），使得添加后的点集是中心对称的。问有多少个不同的点可能成为对称中心，并将他们的坐标输出。

数据范围：$n \leq 2 * 10^5, k \leq 10$

难度：★★

8.5.2 Analysis

算法：枚举

分析：如果一个点集是中心对称的，我们对整个点集进行水平序排序（x为第一关键字，y为第二关键字）。可以想象：第一个点和最后一个点关于中心对称，第二个点和倒数第二个点关于中心对称，以此类推。我们称这种中心对称的点为配对点。

有了以上一个结论，我们先将给定点集进行水平排序，问题变成添加K个点，使得上述配对点拥有同样的中心。

由于K非常小，我们实际上可以枚举第一对配对点的位置，再进行检验即可。也就是枚举配对点(l,r)，使得[1,l-1]通过添加点配对，[r+1,n]也通过添加点配对，然后检验l到r之间是否能用剩下的可添加点帮助配对即可。

复杂度$O(n * k^2)$。

提示与补充：注意$K > n$的情况。

8.6 147B Smile House

8.6.1 Description

题意：在一个n个点m条边的有向带权图中，找一个权值和为正的环，并且经过的点数最少。

数据范围：$n \leq 300, m \leq 90000$

难度：★★★

8.6.2 Analysis

算法：倍增

分析：有别于一般的最短路，最小环问题，本题提出了两个要求：权值和为正，并且环的长度尽可能短。

考虑我们能否快速得到在走不超过k步的情况下，能够走出的最大环。

这样一个问题应该是经典的，定义f[k][a][b]表示走不超过k步，从a到b的最大路径权值和。f[k][a][a] 即为我们想知道的答案。并且从f[k]推到f[k+1]也是容易的。于是f[k]我们可以用倍增得到。复杂度为$O(n^3 logk)$。

所以我们可以二分答案k，判断最大环是否已经大于0。可以预处理$f[2^t]$，将整个问题的复杂度降为$O(n^3 logn)$，足以通过全部数据。

提示与补充：在两个要求的情况，一般是以某一维为下标，另外一维为值进行求解。

8.7 217E Alien DNA

8.7.1 Description

题意：定义在字符串的一个激活操作。每次操作时给定[L,R]，操作会首先将[L,R]复制出来，并且将偶数位放到奇数位前，接着把这个复制出来的串接在原串[L,R]后面。

给初始字符串进行N次激活操作，求最后结果的前k位

数据范围：初始长度，$K <= 10^6$，$N <= 5000$。

难度：★★★

8.7.2 Analysis

算法：数据结构

分析：我们从后往前思考这个问题：最后一次操作完后，我们将保留k个字符。

倒数第二次操作完以后呢？实际上只需要保留k-(r[i]-l[i])即可。也就是说我们可以知道每次操作完成后所需要保留的字符个数。

知道这个信息后，我们只需要每次用平衡树暴力复制即可，复杂度O(klogk)。

本题亦可改成线段树进行维护。似乎存在$O(n^2+k)$的算法，尚待探究。

提示与补充：无

8.8 152D Frames

8.8.1 Description

题意：给定一个$N*M$的字符串方格，每个格子要么是1要么是0，现要你在此方格中找出两个用1组成的矩形，边与网格边平行，且不小于3。使得这两个矩形组合起来就是原网格的1组成的样子。矩形可以相交和重合。

数据范围：$n, m \leq 1000$

难度：★★

8.8.2 Analysis

算法：枚举

分析：由于只需要找到两个矩形，感觉是一个挺水的题，但是一开始没有找到思路，觉得总是情况考虑的不周全=。

其实本题只要找到一个最重要的性质：两个矩形的边一定在行的最大值，次大值，最小值，次小值，列的最大值，次大值，最小值，次小值中产生即可。

于是暴力枚举组合情况，并且在O(n*m)的时间内进行检验即可。

提示与补充：无

8.9 135E Subsequence

8.9.1 Description

题意：定义，如果a串是s串的次连续子串，条件是：
1、a串是s串的子串

2、a串同时是s串的子序列，且这个子序列不能是子串（必须有一处不连续）

现在字符集的大小是K，求有多少个不同的字符串，它的最长次连续子串长度为w。输出结果mod 1000000007。

数据范围：$K <= 10^6, 2 <= W <= 10^9$
难度：★★

8.9.2 Analysis

算法：组合数学
分析：让我们首先探究一下最长次连续子串的性质：

如果最长次连续子串为[l,r]，我们找到它的子序列的开头和结尾l1,r1，可得l1¡l或者r1¿r，否则就会相同。如果l1¡l，则r可以延伸到整个串的末尾，如果r1¿r,l可以延伸到1。所以最长次连续子串一定是前缀或者后缀。

现在要求最长次连续子串长度恰好为w的方案数，我们改成求$<= w$的方案数，最后用w的答案-(w-1)的答案即可。

设串的长度为len，要使答案长度$<= w$，则要使末尾的len-w个字符互不相等且开头len-w个字符互不相等。我们分情况讨论(P为排列)：
1、当$len <= 2w$时，$ans = P(k, len - w)^2 * k^{2w-len}$

2、当$len > 2w$时，$ans = P(k, len - 2 * w) * P(k - len + 2w, w)^2$

枚举len，进行求解即可。

提示与补充：可以先预处理出阶乘mod$(10^9 + 7)$的值。

8.10 183D T-shirt

8.10.1 Description

题意：现在有n个人m种物品，你的任务是选择n件物品（每种物品若干件，可以不选），将这些物品送给n个人。但是你不知道每个人喜欢什么样的物品，只知道每个人接受每种物品的概率分别是什么，你希望找到一种选择物品的方案使得接受人数的期望尽可能大。
数据范围： $n \le 3000, m \le 300$
难度：★★★

8.10.2 Analysis

算法：概率DP+优化
分析：我们首先来考虑某种物品选择x件时接受的人数的期望。设该种物品为A类物品，P[i]代表第i 个人接受A物品的概率，g[i][j]代表前j个人中恰好接受了i个A类物品的概率。
则有DP方程(考虑第i个人选不选择A物品):

$$g[i][j] = g[i-1][j-1] * p[i] + g[i][j-1] * (1 - p[i]) \tag{8.1}$$

那么期望的人数如何计算呢？(如果人数比x少，则对答案贡献i，否则贡献x)

$$E(x) = \sum_{i=0}^{x} g[i][n] * i + \sum_{i=x+1}^{n} g[i][n] * x \tag{8.2}$$

由于各类物品是分别独立的，我们可以分别计算各类物品的期望值，再通过背包合并。这样的复杂度是O(n*n*m)，不能通过全部的数据。于是我们需要进一步的优化，优化空间在哪里呢？通过观察，我们发现E(x)随着x的增大单调递增，并且增量单调递减。这样的性质说明什么？说明了某类物品我们选择的越多，它能给我们带来的增加的收益就越小。所以如果我们现在要增加1件物品，那么我们自然要选择当前能够使答案增加的尽可能大的那件物品，因为选择其他的物品不管是在现在还是将来，所带来的收益的增加都不会比它大了。于是，一个贪心的算法呼之欲出：我们用堆维护当前增益的最大值，每次DP出某类物品数量增加1的收益。复杂度O(mlogm + n * n)，可以轻松AC此题。
提示与补充：实现上有一个小技巧：就是每类物品的g数组可以滚动，这样便不需要动态申请空间。本题用到的增量优化（导函数）非常经典，类似的思想还有2012NOI骑行川藏，推荐去做做（我当时被卡精度了=。。不堪回首。。）

Chapter 9

codeforces(9)

9.1 163D Large Refrigerator

9.1.1 Description

题意：给定长方体体积V，找到边长a,b,c，使得a*b*c=V，且表面积S最小。

数据范围：T组数据，$T <= 500, V <= 10^18$

难度：★★★

9.1.2 Analysis

算法：搜索

分析：先枚举a，a不超过$v^{1/3}$，然后再枚举b，b为V/a的约数，不超过sqrt(V/a)。

然后加上最优化剪枝：就是确定a以后，判断b = sqrt(V/a)时是否比最优解优，如果不优的话就没有必要搜b了。

由于最后答案应该是在a，b,c都比较接近的位置，所以我们把a从大到小枚举，就能很快找到最优解，这应该就是搜索为什么出得来的原因吧。

一道不知道怎么描述的题目= 。

提示与补充：无

9.2 167E Wizards and Bets

9.2.1 Description

题意：给定一个左右部都有K个点的二分图，对于它每一种完美匹配，记mth[i]表示右部点i匹配的左部点是哪个，一个完美匹配的数对(i,j)是逆序对，条件是$i < j$且$mth[i] > mth[j]$。对于二分图每一种完美匹配，如果逆序对数是偶数，则把计数器+1，否则把计数器-1。最后请输出计数器mod素数p的值。

数据范围：$K <= 600, P <= 10^9 + 7$。
难度：★★★

9.2.2 Analysis

算法：行列式
分析：amazing！
首先我们考察相交的路径对答案的影响，可以发现它是不会影响的，因为一定可以交换两个相交的点而改变排列奇偶性，从而抵消。然后就变成了这样一个问题：枚举每一个排列p，计算cnt[i][pi]【左边i号点到右边pi号点的路径方案数】的乘积，+1和-1要看逆序对的个数。然后你会惊恐的发现他就是行列式的定义式，高斯消元在$O(n^3)$的时间解决。

提示与补充：无

9.3 232D Fence

9.3.1 Description

题意：给定n块木板，每块木板的高度为hi，现在选出两段互不相交的长度相等的连续木板，使得对应位置的高度相加正好相等，称这两段木板为互相匹配的。现在有m个询问，每次询问有多少段连续木板与当前的[l,r]这一段木板匹配。

数据范围：$n \leq 10^5, m \leq 10^5$

难度：★★★★

9.3.2 Analysis

算法：后缀数组，数据结构

分析：设两段连续木板分别为[l1,r1]，[l2,r2]。则满足h[l1] + h[l2] = h[l1+i] + h[l2+i]（0¡=i¡r2-l2）。

对上式做一些变形得：h[l1] - h[l1+1] = h[l2+1] - h[l2]，也就是说它们相邻两项的差互为相反数。所以我们可以将相邻高度做差再取反以后，就变成了匹配问题。

做出该串的后缀数组之后，现在的问题变成怎么判断不相交。互不相交，即l2¿r1或者r2¡l1，也就说每次询问时，我们得判断一个区间内¡某个权值的元素有多少，对于这个问题，只需要离线排序后用树状数组维护即可。

复杂度$O(nlogn)$，是不折不扣的代码题=。

提示与补充：无

9.4 175E Power Defence

9.4.1 Description

题意：Main Villain以1米每秒的速度从(-oo,0) 运动到(+oo,0)。在所有点(x,1) 和(x,-1)上Vasya可以修塔，其中x是整数。有三种类型的塔：火、电、冰。不过，同一个点上只能修一个塔。每种类型的塔有一个固定的攻击半径和每秒伤害（除了冰塔）。如果某一时刻Main Villain在k座冰塔的攻击范围内，他的速度将被减少至$1/(k+1)$。现在每种塔的数目已知，Vasya希望知道能对Main Villain造成最大伤害的塔的布置方案。所有距离以米为单位给出。可以将Main Villain和塔近似看做平面上的点。Main Villain在一座塔的攻击范围内是指他与塔的欧几里得距离小于等于塔的攻击半径。

数据范围：$0 <= nf, ne, ns <= 20, 1 <= nf + ne + ns <= 20 1 <= rf, re, rs <= 1000 1 <= df, de <= 1000$

难度：★★★

9.4.2 Analysis

算法：贪心

分析：塔的功能是分类的，冰塔负责减速，火塔，电塔负责攻击。一个冰塔的实质是将它所覆盖的那一段的所有攻击*2。

结论：塔与塔之间不会有空隙。可以通过调整法证明，形象的理解是因为塔与塔尽量挨的拢些，总不会变差。

所以我们只需要枚举20个位置中哪些塔是冰塔，接着算出剩余位置的价值，从大到小进行选择即可。

复杂度O(C(20, 10) * 20)。

提示与补充：无

9.5 176D Hyper string

9.5.1 Description

题意：给定一些基本串，A串有一些基本串（可重复，最多2000个）组成，再给定B串，求A、B 串的最长公共子序列。

数据范围：基本串数<= 2000，基本串长度和<= 10^6，B串长度<= 2000。

难度：★★★

9.5.2 Analysis

算法：DP

分析：求最长公共子序列的一般做法是定义f[i][j]=k代表A串取前i个，B串取前j 个，能够得到的最长公共子序列是多少。

将方程变为f[i][j][k]=1，我们对此方程做常用的变形，令f[j][k]=最小的i。即B串取前j 个，最长公共子序列为k时，A串最少取多少个。

转移时需要快速找到A串i位置后的第一个特定字符在哪里，只需倒序扫描预处理一遍即可。

复杂度：$O(L + n^2 * 26)$。

提示与补充：无

9.6 178F2 Representative

9.6.1 Description

题意： 定义f(x,y)表示字符串x和y的最长公共前缀的长度，给定一个字符串集合，从其中选取k个，使得k个字符串两两求f值求和得到的值尽可能大，输出这个最大值。

数据范围： $1 <= n <= 2000$,字符串最大长度为500。

难度： ★★

9.6.2 Analysis

算法： 分治

分析： 首先将所有的字符串按字典序进行排序。再求出它们每两个相邻的字符串的最长公共前缀，定义为h[i]。

这样，任意两个字符串i，j的最长公共前缀就等于i到j的h函数的最小值。

然后找到h函数中的最小值v，将h数组分为两部分，分别递归求出取x个的最优值，再进行合并，由于v是最小值，所以合并时的最长公共前缀一定是v，设左边部分的长度为l,右边部分长度为r,可以在O(l*r)的时间内进行合并。

调用上述函数，直到不用递归为止。最坏情况下复杂度$O(n^2)$。

提示与补充： 关于最坏情况复杂度可以用函数f[n] = max(f[x]+f[y]+x*y)定义，可以证明f[n] $= n^2/2$。

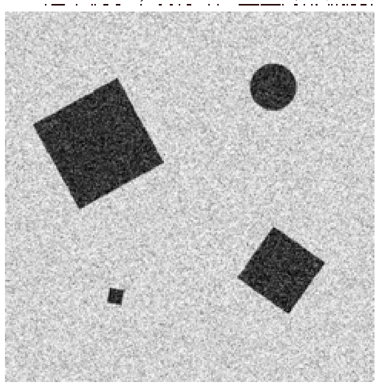
9.7 178E3 The Beaver's Problem II

9.7.1 Description

题意：给定一个n*n的点阵图，就是说，一个矩阵中每个点有自己的颜色，而且矩阵的大小和图的大小相同。

图中的白色代表背景。同时，这个图包含一些黑色的规则形状。已知这个图只包含两种形状：正方形和圆。你的任务是计算给定图中圆的个数和正方形的个数。

图中的正方形可以任意旋转。另外，这个图也有可能包含一些噪音在周围。也就是说，原来的图中每个点有0.2的概率变成相反的颜色。



数据范围：$1000 \leq n \leq 2000$

矩阵中每个数为0或1，保证正方形的边长和圆的直径大于15像素点，任意两个图形间距至少10像素点。保证图形个数不超过50个。保证一个人可以简单的计数图中的圆和正方形的个数。

难度：★★★★

9.7.2 Analysis

算法：图像识别

分析：0.2的概率反色的条件真是很凶残很凶残。。考虑首先对整个点阵图进行“模糊”，模糊的方式是考虑一个点周围8个点的颜色，如果不同颜色比相同颜色多，则对该格子反色。

对该图做了若干次模糊后(我做了5次)。我们可以假定这个图已经较为接近原图了。

这个时候把所有的联通块找出来（我选择的四联通块）。如果块的大小比较小（我选择的< 70）就认为它是区别于规则图形的杂质，直接删除。现在我们得到了很多个联通块，然后要做的就是本题的核心问题：如何判断它是圆还是正方形？

1、面积法：我们找到该图形的对称中心，计算最远点到它的距离(设为r)，然后计算面积（点的个数）与r^2的比，如果更接近圆，则看成圆，否则看成矩形。

2、其实我个人认为面积法是不太靠谱的，主要是面积不一定可以还原的出来。仔细思考，圆区别于正方形最大的一点在于圆是光滑的，而正方形有棱角。所以我想可以尝试统计离对称中心最远的若干点的距离的关系，如果这些距离相差不大，则认为是圆，如果相差较大，则认为是矩形。不过这个方法我没有尝试，仅供参考=。

感觉图像识别的题目还是很有趣，不过调试参数略显蛋疼。

提示与补充：orz 人脸识别！！

9.8 185D Visit of the Great

9.8.1 Description

题意： $LCM(k^{2^l}+1, k^{2^{l+1}}+1, k^{2^{l+2}}+1, ..., k^{2^r}+1)$ mod p.

数据范围： T组数据$\leq 10^5$, $1 \leq ki \leq 10^6, 0 \leq li \leq ri \leq 10^{18}, 2 \leq pi \leq 10^9$

难度： ★★★

9.8.2 Analysis

算法： 数论

分析： 假设$x = k^{2^l}+1$,那么$k^{2^{l+1}}+1 = (x-1)^2+1 = x^2-2^x+2$,而$gcd(x, x^2-2^x+1) = gcd(x, 2)$，所以$gcd(k^{2^l}+1, k^{2^{l+1}}+1)$要么为1，要么为2。显然就是k的奇偶问题。

知道了GCD，LCM就好求了。总的来说，结果大致可以归结为：

1.若k是奇数，答案就是$\frac{(k^{2^l}+1)*(k^{2^{l+1}}+1)*...*(k^{2^r}+1)}{2^{r-l}}$

2.若k是偶数，答案就是$(k^{2^l}+1)*(k^{2^{l+1}}+1)*...*(k^{2^r}+1)$

第二个关键在于怎么求$(k^{2^l}+1)*(k^{2^{l+1}}+1)*...*(k^{2^r}+1)$ 其实把这条式化开整理一下：$(k^{2^l}+1)*(k^{2^{l+1}}+1)*...*(k^{2^r}+1) = 1+k^{2^l}+k^{2^{2^l}}+k^{2^{3*(2^l)}}+...+k^{2^{(2^{r-l+1}-1)*(2^l)}}$ =右边就是一个等比公式，直接用公式代进去就可以了

公式有除法运算，求余的话要注意有可能跟p不是互素，不过很容易想到当且仅当$p|(k^{2^l}-1)$是才出现这种情况，特殊处理一下就行了。最后如果k是奇数在除以一下2^{r-l}即可。

提示与补充： 无

9.9 187D BRT Contract

9.9.1 Description

题意：有N个红绿灯，告诉你他们之间的距离l[i]；在0时刻，他们都是绿灯，在g 秒之后变成红灯，在r秒之后又变成绿灯，依次反复……有Q个询问，每次询问在x时刻有个车从第一个红绿灯出发，问到达N号红绿灯的时刻。

数据范围：$N <= 10^5$
难度：★★★

9.9.2 Analysis

算法：数据结构
分析：首先预处理f[i]表示0时刻从i号红绿灯出发到达终点要多少时间，方法是找到之后第一个需要停留的红绿灯k，f[i]=dis(i..k)+f[k]。

然后对于每次询问，找到1号之后第一个需要停留的红绿灯k，ans=dis(1..k)+f[k]。

两部分都需要找从某个点出发第一个需要停留的红绿灯k。即需要找到在模(g+r)意义下，dis[i..k]¿=g 的最小k。可以一开始对dis的后缀和进行离散，然后从后往前预处理，用线段树维护模意义下某段离散区间的最小值即可。复杂度：O(NlogN)。

提示与补充：无

9.10 180B Divisibility Rules

9.10.1 Description

题意： 检查一个数能否被2,4,5,8,10整除的时候只需要检查它的最后一位或几位是否满足某个条件。现把这种规则称为2类型规则。

检查一个数能否被给定的数整除意味着计算这个数的各个数位上的数字之和并判断这个和能否被给定的数整除，那么Vasya称这个规则为3类型规则。

如果我们需要求出一个数的奇数位上的数字之和与偶数位上的数字之和的差值去检查这个差值能否被给定数整除，那么这个规则被称为11类型规则。

把除数分解成一些因数然后检查是否满足一些不同类型的规则，这样的是6类型规则。

剩余的被称为7类型规则。

给定b,d，分别表示进制数和除数。要求输出是哪一种规则。

数据范围： $2 <= b, d <= 100$

难度： ★★★

9.10.2 Analysis

算法： 数论

分析： 2类型只需判断b^k是否是d的倍数。3类型只需判断$b \equiv k = 1$。11类型只需判断$b \equiv k = -1$。6类型需要判断它的因子是否满足2,3,11类型。而7类型则是剩下的情况。

提示与补充： 一段简洁而又明了的神奇代码如下：

```
while (gcd(n, m) > 1) m /= gcd(n, m), k++;
if (m == 1)
{
    printf("2-type\n%d", k);
    goto die;
}
if (k == 0 && n % m == 1)
{
    printf("3-type\n");
    goto die;
}
if (k == 0 && n % m == m - 1)
{
    printf("11-type\n");
    goto die;
}
if ((n * n - 1) / (n % 2 + 1) % m == 0)
{
    printf("6-type\n");
    goto die;
}
printf("7-type\n");
die:;
```

Chapter 10

codeforces(10)

10.1 198E Gripping Story

10.1.1 Description

题意： 平面内有N个磁铁，已知他们的坐标，引力，引力半径，质量；一个磁铁能被另一个磁铁吸引的条件是，在引力范围内，且质量小于等于引力。

现一人固定在某点，初始时拥有一个磁铁，问最多能吸到多少个。

数据范围： $N <= 250000$
难度： ★★★

10.1.2 Analysis

算法： 数据结构
分析： 如果A可以吸引B，则A连向B一条边。即问从起点能够到达多少个点。由于边数是$O(n^2)$级别的，所以不能建出所有的边。我们考虑动态建图。

在BFS过程中，对于当前已经拥有的磁铁，找到一个最有可能被吸引的磁铁，即重力比引力小的所有磁铁中，距离最近的磁铁。

寻找的方法是对重力和引力建立一颗线段树，维护的一段区间内dis最小的点，每次查找在引力范围内dis最小的点，判断是否在半径内即可。复杂度：O(NlogN)。

提示与补充： 无

10.2 176E Archaeology

10.2.1 Description

题意：给定一棵n个树以及边上的权值，进行m次操作，每次操作可以往点集里插入一个点或删除一个点，查询将当前点集里面所有点连接起来的花费。

数据范围： $n \leq 10^5, q \leq 10^5$
难度： ★★★

10.2.2 Analysis

算法： dfs序
分析： 预处理出这棵树的dfs序以及根到每个点的距离，并由此做ST,便于查找两个点的最近公共祖先lca。

那么插入/删除点u时，找到插入点在dfs序中左、右侧的第一个值，记为l,r。动态的维护此时的答案的变化量。

因为l,r在树中是最接近u的，可以套用ST求出两者的lca来得到插入/删除点u点造成多大的贡献。

对l,r不存在的情况进行讨论。。复杂度：o(mlogn)

提示与补充：代码稍有繁琐，小心调试。

10.3 196D Good String

10.3.1 Description

题意： 给一个仅由小写字母组成的字符串S和一个正整数m，要求一个长度与S相同的仅由小写字母组成的字符串S1。

满足以下要求

1.S1的字典序大于S

2.S1不包含长度大于等于m的回文子串

数据范围： $len(S) \leq 4 * 10^5, m \leq len(S)$

难度： ★★★

10.3.2 Analysis

算法： 贪心

分析： 题目只需要我们构造一个字典序大于S的满足要求的S1即可。

考虑从S的第一位从前往后枚举，每次判断放置的第i位到i-m位或者第i-m-1位是否构成了回文串，如果出现回文串，则改变第i位的值，直到合适为止。

因为根据回文串的性质，如果出现了长度为x的回文串就一定出现了长度x-2的回文串，所以我们只需要判断刚好为m的回文串即可。判回文串可以用hash，复杂度o(n)。

提示与补充： hash时用进制hash即可。

10.4 200E Tractor College

10.4.1 Description

题意： 已知c3,c4,c5,m，求一组(w3,w4,w5)使得下式成立：

$$w3 * c3 + w4 * c4 + w5 * c5 = m(0 <= w3 <= w4 <= w5)$$

并且最小化下式

$$abs(w3 * c3 - w4 * c4) + abs(w4 * c4 - w5 * c5)$$

输出一种方案或者判断没有方案。

数据范围： $c3 + c4 + c5 <= 300$, $M <= 3 * 10^5$
难度： ★

10.4.2 Analysis

算法：
分析： 枚举w4，设tmp=w4*c4。则w3和w5需要为二元一次方程的解：

$$w3 * c3 + w5 * c5 = m(0 <= w3 <= w4 <= w5)$$

g为(c3,c5)的最大公约数，利用扩展欧几里得计算此时w3最小值，并用此更新答案。

再令$w3 * c3 <= tmp$且$w5 * c5 >= tmp$的情况下使w3尽可能大，求出对应值更新答案。

最后令$w3+ = c5/g$,$w5- = c3/g$来更新答案，若答案一直未被更新则认为无解。

提示与补充： 注意无解的情况。

10.5 200A Cinema

10.5.1 Description

题意： 给定一个N*M的网格，有K个操作，每次给定(x,y)，表示想要标记这个点，如果这个点已经被标记了，则找到一个哈密尔顿距离最近的未标记点进行标记，仍然有多少则选择水平序最小的。输出每次实际标记的点坐标。

数据范围： $N <= 2000, M <= 2000, K <= 10^5$

难度： ★★

10.5.2 Analysis

算法： 并查集

分析： 有一个看起来很暴力的算法：首先若N¿M，对网格进行旋转。保证了列数大于sqrt(n*m)。

对于询问(x,y)，从x行开始向上下同时扫描每一行，对于扫描的行i查询该行离(i,y)最近的点（用并查集维护），直到abs(x-i)¿ans当前答案时循环结束。乍看这个算法的复杂度是O(K*N)的。

但经过分析发现，如果i已经循环了t次，则说明有以(x,y)为中心的$O(t^2)$个网格已经被标记，所以t 最多是$O(k^{0.5})$。所以这个算法的复杂度是$O(k^{1.5})$。

实现时需要注意常数。（并查集非递归等）

提示与补充： 无

10.6 201D Brand

10.6.1 Description

题意：有N个单词组成的一句话s0，以及M个由若干单词组成的句子si，若s0某个排列是si的子序列，则称s0与si相似。定义两者的差异度为：s0原排列与当前排列中相对位置发生改变的单词对数的最小值。求M个句子中差异度最小的句子。

数据范围：$N <= 15, M <= 10$,所有句子总单词数$L <= 500000$。
难度：★★★

10.6.2 Analysis

算法：DP
分析：显然M个句子每个句子都是独立的，我们只需要求对于每个句子的最小差异度。

对于每个句子，首先我们可以将不是那N个单词的其余单词删除。然后用$F[s(0..2^N - 1)][i(0..n*(n-1))/2]$表示在子序列中已经有s二进制表示的集合这些单词，他们的差异度为i，这个状态的末尾最前可能是哪个位置。

转移时枚举不在s集合中的单词p，然后求next[F[s][i]][p]表示在F[s][i]之后第一次出现p 单词的位置，用来找到下一个状态，而差异度也很容易求得。

Next数组容易在O(L*N)的时间求出。复杂度：$O(M*2^N*N*N+L*N)$。

提示与补充：无

10.7 201E Organization

10.7.1 Description

题意： 有一个长度为n的排列A,你想通过一些询问知道它是什么样的.

每次你构造一个长度为k($0 < k <= m$)的序列B,满足$1 <= Bi <= n$且B中没有相同的元素,系统会根据序列B生成一个长度为k的序列C，Ci的值为j(Aj=Bi).

然后系统随机洗牌C序列后返回给你,问至少要多少次询问才能知道A究竟是什么.

数据范围： T数据组数≤ 1000，$n, m \leq 10^9$
难度： ★★★

10.7.2 Analysis

算法： 组合，贪心
分析： 我们将每次询问看成一个长度为n的0,1串，被询问的为1，没有被询问的为0。设询问了k次，则得到的是一个k*n的矩形：

	1	2	3	4	5	6	7
一	0	1	0	1	0	1	1
二	1	1	0	0	1	0	1
三	1	0	1	1	0	0	0
四	0	1	0	0	0	0	1
五	1	0	1	1	0	0	0

每行的1的个数不能超过m个。什么时候能得到准确的排列A呢，当且仅当每一列的01字符串两两不一样时。

因为我们每次询问，相当于把一些集合进行拆分，如果01字符串不一样，一定被拆到了不同的集合。最后独立出n个集合，也就得到了答案。

我们二分行数k，当确定行数时，最多能扩展出多少满足条件的列呢？

即最多能得到多少个长度为k的不相同的字符串，满足每一行有不超过m个1呢？

我们将条件放松一下：每一行不超过m个1变为总共1的个数不超过k*m。

这样一个问题，我们贪心解决，先考虑没有1的字符串，然后是1个1,2个1,3个1。。直到1的个数超过了k*m。

可以通过调整证明这个问题和原问题等价，即总数不超过k*m的情况总可以调整成每一行不超过m。

于是本题完美解决了。复杂度O(T*log*log)。

提示与补充：需要预处理组合数。

10.8 204E Little Elephant

10.8.1 Description

题意：给定一个字符串，长度为N。对于每个字符串，询问有多少子串是所有N个字符串中至少K个的子串。

数据范围：$N, K <= 10^5$。字符串总长$<= 10^5$。
难度：★★★

10.8.2 Analysis

算法：后缀数组+单调队列
分析：首先将所有字符串拼在一起进行后缀排序。按顺序枚举每个后缀，需要计算的是它有多少个前缀至少被K个字符串包含。

可以维护对于每个起点i，向右扩展到哪里就会包含K个字符串。对于这样的一段区间，公共前缀是height中最小值（定义为区间权值）。

而要计算某个后缀i的最长至少被K个字符串包含的前缀个数，则是要求包含i位置的所有区间权值最大值。这里可以用单调队列维护。

而注意到区间仅是包含K个字符串的最小区间，可能可以向后继续扩展，则i位置最长公共K- 前缀可以由最后一个不包含i的区间扩展到i之后更新。这里也可以用单调队列维护。

总复杂度：O(nlogn+N)。

提示与补充：细节较为繁琐。

10.9 207B Military Trainings

10.9.1 Description

题意：最初的时候，n辆坦克排成一列，从头到尾依次编号1到n。在整个练习期间，恰好有n条信息从排头的坦克传到最后面的坦克。每当一次信息传完后，最后的坦克就会移动到最前方，进行新一轮信息的传输。2个坦克信息间的传输花费1s，且n个坦克分别有ai的接受半径，询问n轮信息最少花多少时间。

数据范围：$n \leq 250000$。

难度：★★★

10.9.2 Analysis

算法：倍增

分析：我们考虑将过程反过来，改为从后往前传递信息。这样每次会选择最靠前的一辆坦克进行传递。我们对这种传递建立有向边，那么第i次询问变成要往前多少次才能跳过n辆坦克。

对于这样一个问题，惊讶的发现倍增就可以解决了。记录往前每辆坦克往前2^k步所能到达的地方，每次查询类似于二分答案的查询即可。复杂度O(nlogn)。

提示与补充：无

10.10 207A Beaver's Calculator

10.10.1 Description

题意：给定n行，每一行给定ai个数，把这n行数合并，要求满足对于原先某一行的数，在新的序列里相对顺序是固定的（和原先一样），并且设新序列中相邻的2个数，前一个数大于后一个数的情况为“坏对”，使“坏对”尽可能少，输出“坏对”的最少出现次数和方案。

保证$\sum ai \leq 2*10^5$。

数据范围：$n \leq 5000, ai \leq 5000$
难度：★★

10.10.2 Analysis

算法：贪心
分析：“坏对”的最小出现次数是很显然的，只要将每一行的坏对数计算出来取一个最大值即可。

如何构造方案呢？用tmp表示当前最优序列中最后一个元素，用一个set存下当前每一行第一个数（即可选择的数）。

每次从set查找$>= tmp$的最小的数，将它加入最优序列，并且将它所在行新的可加入元素加入set。

如果找不到$>= tmp$的数，则把最小的数加入最优序列，并且让答案+1。

复杂度O(200000*log200000)。

提示与补充：无

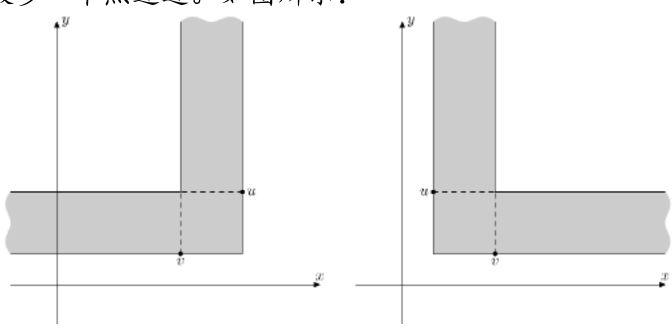
Chapter 11

codeforces(11)

11.1 167D Wizards and Roads

11.1.1 Description

题意： 平面内给出随机生成的N个不同的点，现在需要给这些点连边。可以连边的点对(u,v)（$Yu > Yv$）需要满足，对于在其形成的拐角内任意点w，存在一点s在u与w之间，不在拐角内，且$Ys > Yv$。规定每个点只能和最多一个点连边。如图所示：



有M个询问，每次询问横坐标为[L,R]内的点最多可以连多少条边。

数据范围： $N, M <= 10^5$
难度： ★★★★

11.1.2 Analysis

算法： 笛卡尔树
分析： 经过分析，可以发现连边(u,v) (Yu¿Yv)，U的左边只有一个满足条件的v，右边也只有一个满足条件的v。具体地，可以与U相连的V，一

定是U与左边第一个比它高的点之间，最高的点；右边同理。不难发现这是一棵树（笛卡尔树），我们可以在树上进行DP。F[i][0..1]表示DP 完i子树，i是否连边时候的状态。

有多组询问，我们可以将这颗树建出来，每次查询时可以在期望O(logn)时间内解决。（因为数据是随机的）。

预处理建树+log查询。复杂度：O(NlogN+MlogN)。

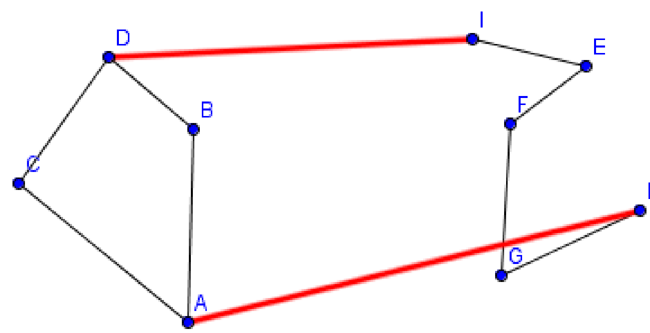
提示与补充： 这样一棵树也叫笛卡尔树，想了解更详细内容的可以wiki。

11.2 209C Trails and Glades

11.2.1 Description

题意：给定一个N个点M条边的无向图，问最少添加多少条边，能够使得该图存在一条经过1号点的欧拉回路。

下图是一个例子：



数据范围：$N, M <= 10^6$。
难度：★★

11.2.2 Analysis

算法：贪心
分析：存在欧拉回路的条件是：连通且每个点度数都为偶数。

如果图已经连通，奇点个数为x，那么答案为x/2。我们先将连通块求出来，并求出每个联通块有多少个奇点；要做的就是先让这个图变为连通图

且尽量让连通后的奇点更少。不难发现一个贪心的策略：每次选出两个奇点最多的连通块合并。我们用堆来维护这个贪心算法。复杂度：O(nlogn)。

提示与补充：本题亦存在不用堆的O(n+m)贪心做法。

11.3 212B Good Substrings

11.3.1 Description

题意：给定一个只有小写字母的字符串。给定M个询问，每次给定一个小写字母集合，问字符串有多少个极大子串包含且仅包含这个集合中的字母。

数据范围：字符串长度$N <= 10^6$，询问数$M <= 10^4$。
难度：★★★

11.3.2 Analysis

算法：字符串处理
分析：注意到问题要求的是极大子串，极大子串不存在包含关系，所以最多只有26*N个。我们从子串长度从小到大来求，每次在上一次基础上向后增加一个新字符。

注意到询问只有10000个，大部分集合都是无效的。我们可以先将有效的集合hash记录。每找到一个新的极大子串，将答案加入对应的集合中。

复杂度：O(26*N)。

提示与补充：无

11.4 212D Cutting a Fence

11.4.1 Description

题意：给定一个长度为N的序列，每次询问在这个序列中选出长度为k的一段，这段的最小值期望是多少。
数据范围：$N <= 10^6$，询问数$<= 10^6$。
难度：★★★

11.4.2 Analysis

算法：数据结构
分析：对于询问K，我们知道总共有(N-k+1)段长度为K的区间，如果我们能够求出每一段最小值的和s，那么ans=s/(n-k+1)。

我们对于其中每一个元素，求出左边第一个比它小和右边第一个比它小的位置，那么在中间的，并且跨过该元素的段的最小值一定是这个元素。

我们考虑它对答案的影响，设它左边有L个右边有R个比它小($L > R$)，不难发现这个元素对答案的影响就是k属于[1,R]答案+=k*x；k属于[R+1,L]答案+=R*x；k属于[L+1,L+R]答案+=(L+R-k+1)*x。

可以用线段树维护，也可以通过差分后通过标记+扫描的方法。复杂度:O(NlogN)或者O(N)。

提示与补充：无

11.5 212C Cowboys

11.5.1 Description

题意：有N个人在一个环，指着环中相邻的两个人中的某一个。如果两个人互相指着对方，那么在下一秒他们将同时指向相反的方向。现给定一个每个人的状态，求前一秒可能的状态数。

数据范围：$N <= 100$
难度：★★

11.5.2 Analysis

算法：DP
分析：简单DP。首先将环在某处断开，形成一条链。枚举链头的状态来分别进行DP。

F[i][0]表示前i-1个人已经合法，第i个人指向第i+1个人的方案数。

F[i][1]表示前i个人已经合法，第i个人指向第i-1个人的方案数。

DP完进行讨论后加入答案。复杂度：O(n)

11.6 213E Two Permutations

11.6.1 Description

题意： 给出两个排列A，B，问有多少个整数d满足把A元素全部+d之后，A是B的子序列。

数据范围： 序列长度<= 200000

难度： ★★

11.6.2 Analysis

算法： 数据结构

分析： 首先我们进行一下转换，令a[A[i]]=i,b[B[i]]=i(这是解决这类问题一个常用手段)。

接下来我们会发现问题就是求a数组的“趋势”在b 数组中出现多少次。可以用树状数组维护的KMP匹配解决。也可以利用hash解决。复杂度：O(nlogn)。

提示与补充： 无

11.7 217C Formurosa

11.7.1 Description

题意： 现有N个为0或1的未知数（N个数的值不是全部一样），有一个含’?’的二进制表达式，现在可以任意将未知数带入表达式，你可以得到表达式的结果。问是否可以得到每个未知数具体的值。

数据范围： $N <= 10^6$。表达式长度$<= 10^6$。

难度： ★★★

11.7.2 Analysis

算法： 贪心

分析： 本题的做法基于如下的一个结论：

如果存在一种代入的方法：使得每个'?'要么为x，要么和x不同，并且最后的表达式的值是x 或者x的相反数，我们则找到了求出每个未知数的办法。否则找不到。

说明：我们总可以通过尝试，使得上式出现。

于是我们可以通过定义一个4位二进制f[i]表示i表达式能否计算出0，1，x，x的相反数来解决此问题。递归计算f[i]函数，合并的时候比较繁琐。

如果最后表达式的f值能够计算出x或者x的相反数，则YES，否则NO。

提示与补充：注意可能会栈溢出，复杂度O(Len)。然后N实际上是个傻瓜读入（没有用处）。

11.8 229E Gifts

11.8.1 Description

题意：金鱼有M个袋子，每个袋子里面有k[i]个价值不同的物品；现在要像金鱼索要价值前N大的物品。如果该物品所在袋子当前有多个物品，金鱼将等概率拿一个；如果有多个价值前N大的物品方案，将等概率选择一个进行索取。问能拿到最大价值的概率是多少。

数据范围：$M <= 1000$，$N <= \sum ki <= 1000$。
难度：★★★

11.8.2 Analysis

算法：DP
分析：DP。首先处理哪些袋子索取次数是一定的，哪些是不确定的（因为前N大价值物品方案不同）。

F[i][j]表示前i个袋子，有j个不确定的袋子次数+1，能够取得最优价值的概率。

对第i个进行讨论是否是不确定的袋子，进行背包DP。

答案：F[m][n-b]/c[g][n-b](b表示确定的袋子数，g表示与价值第n大物品相同价值的物品数）。

提示与补充：无

Chapter 12

codeforces(12)

12.1 113D Museum

12.1.1 Description

题意：有一个n个点m条边的无向图，没有重边和自环，2个人分别从房间x,y出发，他们每个人采取如下的行动方法：每一分钟做决定往哪里走，有pi的概率在这分钟内不去其他地方（即呆在房间不动），有1-pi的概率他会在相邻的房间中等可能的选择一间并沿着走廊过去。这里的i指的是当前所在房间的序号。当两个人在某个时刻选择前往同一间房间，那么他们就会在那个房间相遇。他们按照上述方法行动直到他们碰面为止。求两人在每间房间相遇的概率。

数据范围：$1 <= n <= 22$。

难度：★★

12.1.2 Analysis

算法：高斯消元

分析：定义f[a][b]表示甲从x出发，走到a，乙从y出发，走到b，中间不相遇的概率。定义d[a]为a所连的边数。

容易得出：

$$f[a][b] = \sum_{u \neq v} f[u][v] * (1 - pu) * (1 - pv) * \frac{1}{d[u]} * \frac{1}{d[v]}$$

$$+ \sum_{a \neq v} f[a][v] * pa * (1 - pv) * \frac{1}{d[v]}$$

$$+\sum_{u\neq b} f[u][b]*(1-pu)*pb*\frac{1}{d[u]}$$

$$+f[a][b]*pa*pb$$

由上式我们可以得到一个等式。注意f[x][y]的右边应该再+1，因为这个概率计算的实际上是所有x到a，y到b的路径的概率之和。

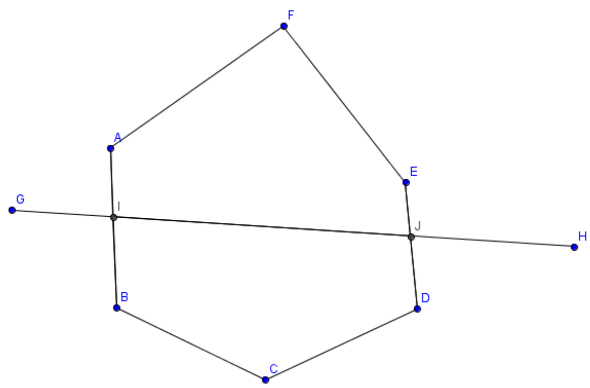
将f[a][b]看成未知数，用高斯消元求解即可。最后的答案只需要枚举a,b走到同一个i即可。

复杂度$O(22^6)$。
提示与补充：高斯消元时小心乘爆。

12.2 75E Ship's Shortest Path

12.2.1 Description

题意：平面上有点s，点t和一个岛屿(其他地方为海洋),定义一个点在安全位置当且仅当它在s,t两地之间的线段或者岛屿的边界上，在海洋上行动1单位距离花费为1，岛屿上为花费2/单位距离，保证岛屿是一个凸多边形，不会出现三点共线，且s,t不会在岛屿内部或边界。求最小花费(精确到小数点后6位)。



数据范围：$n <= 100, |x|, |y| <= 100$。
难度：★

12.2.2 Analysis

算法：计算几何
分析：此题非常简单，只需要用线段切割凸多边形即可，类似于半平面交的暴力做法。注意线段与凸多边形的边共线的情况。
提示与补充：无

吐槽：满纸荒唐言，一把辛酸泪，都言作者痴，谁解其中味。