IOI2015中国国家集训队作业
解题报告

绍兴一中 任之洲

Contents

1 Paradox Sort 9

2 Runs 11

3 Letter Stamper 12

4 Candy Store 14

5 Year of More Code Jam 15

6 Min Perimeter 18

7 Wi-fi Towers 20

8 The Year of Code Jam 21

9 Levko and Sets 23

10 Levko and Game 25

11 Xenia and String Problem 26

12 Transferring Pyramid 28

13 Jeff and Removing Periods 30

14 Doodle Jump 31

15 Pumping Stations 33

16 Xenia and Dominoes 35

17 Candies Game 36

18 Three Swaps 37

19 GCD Table 38

20 Optimize! 39

21 Rectangle And Square 41

22 Buy One，Get One Free 42

23 Lucky Tickets 44

24 Theft of Blueprints 45

25 Binary Key 46

26 The Great Julya Calendar 47

27 The Evil Temple and the Moving Rocks 48

28 Reclamation 49

29 The Red Button 50

30 Tournament-graph **51**

31 Two permutations **52**

32 Ciel and Flipboard **53**

33 Have You Ever Heard About the Word? **54**

34 PE lesson **55**

35 Summer Homework **56**

36 Good Substrings **57**

37 Sereja and Squares **58**

38 Fetch the Treasure **59**

39 Biologist **61**

40 Context Advertising **62**

41 Tennis Rackets **63**

42 White, Black and White Again **64**

43 Polygon **65**

44 Olya and Graph **66**

45 Playing with String **67**

46 Random Ranking **68**

47 Yaroslav and Algorithm **70**

48 Yaroslav and Arrangements **72**

49 Greg and Caves **73**

50 Shaass and Painter Robot **74**

51 Distinct Paths **75**

52 Ksusha and Square **76**

53 Close Vertices **77**

54 Polo the Penguin and Lucky Numbers **78**

55 Tourists **79**

56 Ladies' Shop **80**

57 Positions in Permutations **81**

58 Cow Tennis Tournament **82**

59 k-Maximum Subsequence Sum **83**

60 The Last Hole! **84**

61 Dima and Figure **86**

62 Maximum Waterfall **87**

63 Wall Bars **88**

64 Berland Traffic **90**

65 BerDonalds **91**

66 More Queries to Array... **92**

67 Roadside Trees **93**

68 Rhombus **94**

69 Maxim and Increasing Subsequence **96**

70 Maxim and Calculator **97**

71 Dividing Kingdom **98**

72 Little Elephant and Broken Sorting **99**

73 Liars and Serge **100**

74 Rats **101**

75 Printer **103**

76 Two Sets **105**

77 Donkey and Stars **106**

78 Endless Matrix **107**

79 Piglet's Birthday **108**

80 Colorado Potato Beetle **109**

81 Numbers **110**

82 Road Repairs **111**

83 Torcoder **112**

84 Meeting Her **113**

85 Cyclical Quest **114**

86 Number Challenge **115**

87 Figure Eight **116**

88 Photo **117**

89 Hill Walk **118**

90 Cows in a Skyscrape **119**

91 First! **120**

92 Tower of Hay **121**

93 Cleaning Up **122**

94 Cow Neighborhoods **123**

95 Triangle Counting **124**

96 Land Acquisition **126**

97 Toys **127**

98 Fence **128**

99 Best Cow Line **129**

100Cow Patterns **130**

1 Paradox Sort

【题目来源】

GCJ 2014 Final D

【简要题意】

求一个字典序最小的n的排列，排列的第一个元素直接拿入，然后依次判断接下来的数，如果那个数可以替换当前数，那么就替换，要求最后在手上的数是M。

给出的替换关系是一张有向完全图。

共T组数据。

数据范围：$T, n \leq 100$

【解题思路】

由于要求字典序最小，所以考虑逐位确定，我们可以用手上的数和还没有出现过的数描述当前局面，我们需要判断当前局面是否可以使得最后手上的数是M。

下面提供一种判断方法。

设当前点是S，目标点为T，还没有到达过的点集为Q(S不在Q中)，若X可以替换Y那么就连一条X到Y的边。

如果S=T那么这种情况比较特殊，Q中的点都必须不能替换T，否则就不能保证终点为T了。

那么想像以T为起点作一棵生成树，设生成树上的点集为P，P⊆Q，使P最大化。

(1)如果Q中有些点不在P内，那么可以考虑用S来解决，因为S不在生成树中，如果那些点都不能替换S那么也是合法的，否则不合法。

(2)显然P中必须有至少一个点能替换S。

满足上面两条约束的所有局面都是合法的，我们可以构造一种排列来使得终点为T。

把所有点按照生成树中的深度从大到小排序，依次添加，想像这一过程。首先会有一个点替换掉了S，然后因为是按深度排序的，所以在还没出现的点中一定会有点能替换它，当前点的深度会不断减小，最后即为根，也就是T。

这样判断的效率是$O(n^2)$。

时间复杂度$O(Tn^4)$，空间复杂度$O(n^2)$。

2 Runs

【题目来源】

GCJ 2011 Final A

【简要题意】

每一个极大连续相同子序列称为一个“Run”，给出一个长度为n的串，算出对这个串的字符重排后有多少串的Run个数和原串一样。

数据范围：设原串Run个数为m，$m \le 100$，$n \le 450000$

【解题思路】

考虑对每种字符单独转移，设$f[i][j]$表示已经用到第i种字符，已经做出j个Run的方案数。

转移有2种，一种是在原先的两个Run中间插入一个新Run，还有一种的在原先的一个Run中，把这个Run断成两个，再在中间插入一个新Run。第一种转移共有$j+1$个转移位置，第二种转移共有$len-j$个转移位置，可以枚举转移数量，用乘上相应的组合数来计算具体数值。

时间复杂度$O(26m^3+nm)$，空间复杂度$O(nm)$。

3 Letter Stamper

【题目来源】

GCJ 2010 Final A

【简要题意】

给出一个由“A”“B”“C”构成的序列（长度为n），你可以使用一个栈，通过“在栈顶加一个元素”，“删一个栈顶元素”，“打印栈顶”三种操作来打印这个序列，求最小操作次数。

T组数据。

数据范围：$T \leq 20$，$n \leq 2000$

【解题思路】

解决这个问题需要用到如下的性质：

(1)如果下一个要打印元素就在栈顶，那么一定是直接打印。如果栈顶可以直接打印，一定是先打印为优。

(2)栈中不会有两个连续的相同元素，因为这两个元素的作用是一样的。

(3)最优解的栈中可以不出现形如“XYX”的连续序列。假如栈顶是“XY”，我们需要“X”这个元素，我们加入和删除这个“X”使用的操作次数，可以删除并重新加入“Y”。

所以，栈中的元素一定是形如“$XYZXYZXYZ...$”分布的，所以栈中只有6种情况，“ABC”“ACB”“BAC”“BCA”“CAB”“CBA”，所

以DP状态只要记录前3个字符和栈中元素个数，根据当前栈顶字符进行转移，DP数组可以采用滚存。

时间复杂度为$O(Tn^2)$，空间复杂度$O(n)$。

4 Candy Store

【题目来源】

GCJ 2010 Final C

【简要题意】

有一家糖果店，每天会来k个客人，每个客人购买的糖果数会是$[1, C]$间的任意正整数，而糖果店只出售包装好的糖果，包装好的糖果当然是不能拆开零售的，问糖果店至少需要准备多少盒糖果。

T组数据。

数据范围：$T \leq 100$，$k \leq 1000$，$C \leq 10^12$

【解题思路】

设阶段n为所有客人购买的糖果的总和不超过n，且购买得最多的客人不超过$\lceil \frac{n}{k} \rceil$个糖果。那么需要求的就是阶段$kC$的答案。

假设现在在阶段x，那么要到达下一个阶段，一定有一个客人买了价值为$\lceil \frac{x}{k} \rceil$的物品，于是就多准备一个该价值的盒子，那么对于剩下的客人，虽然他们的糖果数的上限会改变，但是改变量是可以由那个已经满足条件的人省下的糖果中取的，所以多取一个盒子可以从阶段x转移到阶段$x + \lfloor \frac{x}{k} \rfloor + 1$，模拟。

时间复杂度$O(T\sqrt{C})$，空间复杂度$O(1)$。

5 Year of More Code Jam

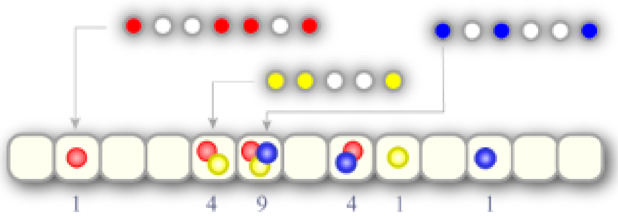
【题目来源】

GCJ 2009 Final A

【简要题意】

有n套比赛，每套比赛有m_i场比赛组成，每场比赛在这套比赛开始后的第$d_{i,j}$天。

这一年总共有T天，每套比赛在每天开始的概率都是相等的，且一定会开始。



若这一天有x场比赛，就会产生x^2的愉悦值，求出这一年的期望愉悦值之和（有些比赛的时间可能会被拖到后一年，这些不需要计算），需要输出分数表示的精确值。

数据范围：$T \le 10^9$，$n, m_i \le 50$，$d_{i,j} \le 10000$

【解题思路】

设E_i为第i天的愉悦值。

$$e(\sum E_i) = \sum e(E_i)$$

设X_i为第i天的比赛数。

$$\sum e(E_i) = \sum e(X_i^2)$$

设$Y_{i,j}$为第i天第j套比赛是否有比赛，有则为1，没有则为0。

$$e(X_i^2) = e((\sum Y_{i,j})^2) = e(\sum Y_{i,j}^2 + \sum_{j=1}^{n} \sum_{k=j+1}^{n} 2 * Y_{i,j} * Y_{i,k})$$

因为$Y_{i,j}$只会是0或1，所以$Y_{i,j}^2 = Y_{i,j}$。

$$\begin{aligned}
& e(\sum Y_{i,j}^2 + \sum_{j=1}^{n} \sum_{k=j+1}^{n} 2 * Y_{i,j} * Y_{i,k}) \\
&= e(\sum Y_{i,j} + \sum_{j=1}^{n} \sum_{k=j+1}^{n} 2 * Y_{i,j} * Y_{i,k}) \\
&= e(\sum Y_{i,j}) + e(\sum_{j=1}^{n} \sum_{k=j+1}^{n} 2 * Y_{i,j} * Y_{i,k}) \\
&= \sum e(Y_{i,j}) + \sum_{j=1}^{n} \sum_{k=j+1}^{n} 2 * e(Y_{i,j} * Y_{i,k})
\end{aligned}$$

设$d_{j,k}$为第j套的第k场。

$$D_{i,j} = \sum_{k|d_{j,k} \le i} 1$$

那么

$$e(Y_{i,j}) = \frac{D_{i,j}}{K}$$

$$e(Y_{i,j} * Y_{i,k}) = \frac{D_{i,j} * D_{i,k}}{K^2}$$

从这里可以看出，当$i > Max(d_{j,k})$时

$$E_i = E_{i-1}$$

那么利用这个结论，可以写出一个DP，已经足够通过全部数据，不过我们可以做的更好。

现在我们需要求的是$\sum_{i=1}^{K} \frac{D_{i,j}}{K}$和$2 * \sum_{i=1}^{K} \frac{D_{i,j} * D_{i,k}}{K^2}$，注意这里已经没有$e()$了。

考虑每个$d_{j,x}$对答案的贡献，即把具体某两套比赛的某两场匹配起来。

$$\sum_{i=1}^{K} \frac{D_{i,j}}{K} = \frac{\sum_{j=1}^{n} \sum_{x=1}^{m_j} (K - d_{j,x} + 1)}{K}$$

$$2 * \sum_{i=1}^{K} \frac{D_{i,j} * D_{i,k}}{K^2} = 2 * \frac{\sum_{j=1}^{n} \sum_{k=j+1}^{n} \sum_{x=1}^{m_j} \sum_{y=1}^{m_k} (K - max(d_{j,x}, d_{k,y}) + 1)}{K^2}$$

这样我们可以设计出一个$O(n^2m^2)$的算法已经足够优秀，对于$(K - max(d_{j,x}, d_{k,y}) + 1)$这一步，也可以利用单调性扫一遍把算法优化到$O(n^2m)$。

时间复杂度$O(n^2m)$，空间复杂度$O(nm)$。

6 Min Perimeter

【题目来源】

GCJ 2009 Final B

【简要题意】

给出一个点集，求出这个点集中3个点能构出的三角形中的周长最小的三角形的周长(三点共线也认为是三角形)。

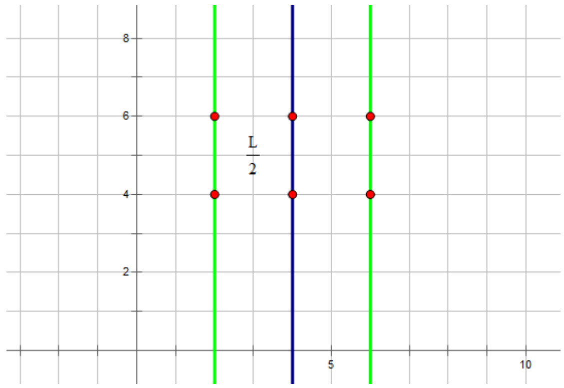
数据范围：$n \leq 10^5$

【解题思路】

经典问题，可以用分治来解决。

先把点集分成左右两部分各$\frac{n}{2}$个点，递归计算出这两个点集的答案，然后考虑计算跨两个集合的答案。

设已知最优解为L，那么跨集合的解的点只会存在于集合分界线左右$\frac{L}{2}$的范围内，对于在这个范围内的点，与它构成的解的另两个点的上下距离也不能超过$\frac{L}{2}$，在划定范围内这样的点最多为6个(考虑不超过答案的最紧密的6个点的排布，见下图)。



那么计算跨集合的答案时只需要对集合分界线左右$\frac{L}{2}$的范围内的点排序后，然后枚举每个点与它上方的6个点暴力计算即可，这里的排序选用归并排序更优。这样复杂度$T(n) = 2T(\frac{n}{2}) + O(n) = O(n \log n)$

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

7 Wi-fi Towers

【题目来源】

GCJ 2009 Final D

【简要题意】

有n个信号塔，每个塔都由坐标(x_i,y_i)表示，且有一个影响半径r_i。

一开始每个信号塔都使用通讯协议A，现在可以选择升级为通信协议B，会有s_i的收益，这个收益不一定是正的，也就是说也可能会亏损，并且假如一个信号塔使用了通讯协议B，那么在它影响半径中的所有塔也必须使用通信协议B（对影响它的塔不作限制）。

求最大收益，共T组数据。

数据范围：$T \leq 55$，$n \leq 500$

【解题思路】

问题可以转化为，选择一个合法集合，使集合权值最大，而且有若干选了一个就必须选另一个的限制。这个是经典的最大权闭合子图问题，可以用网络流建模来解决。一开始先把所有正权加到答案中，然后正权点和源连边，负权点和汇连边，流量限制为$|s_i|$，如果选了u就必须选v那么就连一条u到v的边，流量限制为∞，答案减去最大流。

这样建图点规模$O(n)$，边规模$O(n^2)$，利用$kd\ Tree$理论上可以把边规模优化到$O(n\sqrt{n})$，但是对于该题的数据范围并无大用。

时间复杂度$O(Tn^4)$，空间复杂度$O(n^2)$。

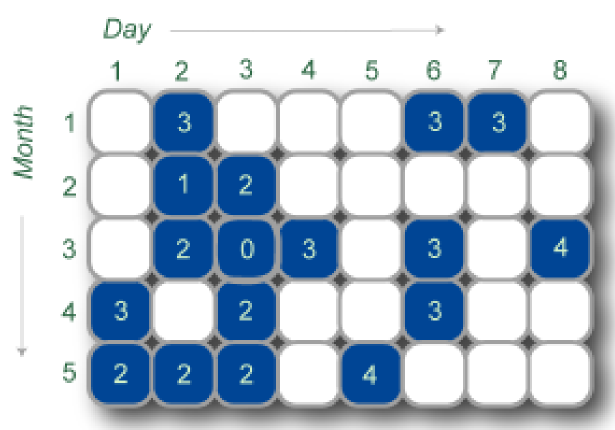
8 The Year of Code Jam

【题目来源】

GCJ 2008 Final E

【简要题意】

一个$n * m$的方格图，每个格子可以选择黑色或白色，一开始每个黑格子都有一个权值4，每个黑格子都会使它四周的黑格子的权值减一。



现在有一些格子的颜色已经确定了，问剩下的格子的颜色确定后最大的权值和是多少。

T组数据。

数据范围：$T \leq 100$，$n, m \leq 50$

【解题思路】

假如新加入一个黑格子，那么它四周每有一个黑格子都会使答案减二，所以它周围最多只能有一个黑格子，不然就可以不放这个格子。

由于方格图是二分图，所以可以用类似最大点独立集的建模来解决。先把所有未确定权值加入答案，然后相邻的格子连边，根据已知颜色的格子来确定新格子的初始权，做最小割。

点规模和边规模都是$O(nm)$。

时间复杂度$O(Tn^3m^3)$，空间复杂度$O(nm)$。

9 Levko and Sets

【题目来源】

Codeforces 360D

【简要题意】

给出一个长度为n的数列A，一个长度为m的数列B，一个质数p。

现在生成了n个集合，第i个集合这样生成：

（1）一开始集合中只有一个元素1。

（2）从集合中选出一个元素c，把$c * {A_i}^{B_j}\ mod\ p$放入当前集合。

（3）不断重复步骤（2）直到没有元素可以再加入集合。

求这n个集合的并集的大小。

数据范围：$n \leq 10^4$，$m \leq 10^5$，$p \leq 10^9$

【解题思路】

首先需要化简一下集合B，设g为质数p的一个原根，那么设$A_i = g^k$，那么${A_i}^{B_j} = g^{k*B_j}$，根据费马小定理，B集合就可以化简为一个数$b = gcd(p-1, B_1, B_2, ..., B_m)$。

现在考虑计算第i个集合的大小，由于循环节一定存在，所以集合大小一定为$p-1$的约数，可以逐一用快速幂尝试计算循环节大小。约数个数是$O(\sqrt{\frac{p}{\log p}})$的，而常底数的幂次计算可以利用积性来处理

最后一个问题就是计算并集的大小了，从原根的角度来考虑，循环节的长度就固定了集合的组成，所以根据集合的大小可以用容斥处理出并集

的大小。

时间复杂度$O(n\sqrt{\frac{p}{\log p}}+\frac{p}{\log p})$，空间复杂度$O(n+m+\frac{p}{\log p})$。

10 Levko and Game

【题目来源】

Codeforces 360E

【简要题意】

一张n个点$m+K$条边的无向图，m条边的权值已固定，K条边的权值可以在l_i～r_i范围内选择。

你在S_1，对手在S_2，要走到终点T，判断在最优策略下的胜负状态，必胜或平局要输出一种边权分配方案。

数据范围：$n, m \le 10^4$，$K \le 100$

【解题思路】

可以得到一个贪心策略，如果一条边对手需要走，那么选择边权r_i，否则选择边权l_i。

可以采用类似最短路的算法来解决，把S_1和S_2同时选作起点，在做最短路时根据当前点上是自己还是对手来选则边权，即记录$dist_i = min(dist(S_1, i), dist(S_2, i))$以及$type_i$为最短路的类型（从$S_1$到达或$S_2$到达或$dist(S_1, i) = dist(S_2, i)$）。

这样做出的最短路可以准确地判断必胜态，但是对于平局的判断有BUG，即把一条自己和对手需要走的边设为l_i后可以造成平局，这个只需要单独特判就行。

时间复杂度$O(m \log n)$，空间复杂度$O(n+m)$。

11 Xenia and String Problem

【题目来源】

Codeforces 356E

【简要题意】

递归定义了一种串叫$Gray$串，$Gray$串的定义是最中间的字符在串中只出现一次，且按中心划分出的两个串相同且都是$Gray$串。

对于给定的串S（串长为n），求出这个串的所有子串中是$Gray$串的子串的长度的平方和。

数据范围：$n \le 10^5$

【解题思路】

首先可以发现$Gray$串的串长在给定数据范围内很少，因为串长可以表示为$f(0) = 1$，$f(x+1) = f(x) * 2 + 1$ $(x \ge 0)$，所以需要考虑的子串长度的数量是$O(\log n)$级别的。

由于每次只能修改一位，所以我们可以先把不修改的答案算出来，然后再枚举修改某一位，计算修改对答案的贡献。

由$Gray$串的对称性可以得出，假如以i为中心，长度为L的串是$Gray$串，那么以i为中心，长度为$\frac{L-1}{2}$的串也是$Gray$串。

预处理答案我们可以枚举$Gray$串的串长和串的中心位置，然后判断左右是否是相同的$Gray$串，且中心字符只出现了一次，并处理出覆盖每个位置的$Gray$串的长度平方和。这部分计算的复杂度为$O(n \log n)$。

现在考虑修改一个字符计算对答案的贡献，首先先减去覆盖到这个字符的所有原先的$Gray$串的贡献，然后以当前位置为中心枚举$Gray$扩展半径，进行暴力递归，假设我们已知$S[l,r]$是$Gray$串，设$length = r - l + 1$，我们需要继续判断$S[l - length - 1, r]$和$S[l, r + length + 1]$。

考虑这样暴力递归判断的复杂度，除了我们修改的位置，串的其他部分必须是$Gray$串，所以计算次数的上界是原串中的$Gray$串的数量，而$Gray$串的数量是$O(n)$级别的(考虑一个长为L的$Gray$串的所有子串)，设字符集大小为S，那么暴力判断的复杂度为$O(S * n)$。

时间复杂度$O(n \log n + Sn)$，空间复杂度$O(Sn)$。

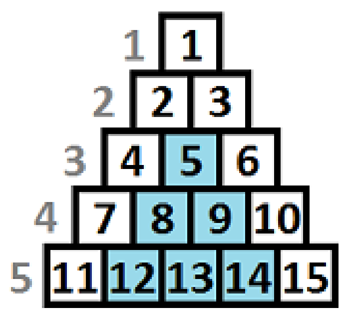
12 Transferring Pyramid

【题目来源】

Codeforces 354D

【简要题意】

给出一个n层的金字塔（金字塔的形态如下图），现在有m个格子需要被覆盖。



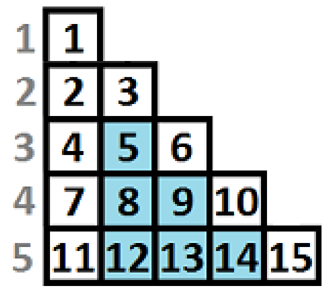
覆盖操作有两种，覆盖一个单独的格子的代价为3，覆盖一个子金字塔（如上图，必须到底）的代价为$size+2$，求覆盖完所有必要位置的最小代价。

数据范围：$n, m \le 10^5$

【解题思路】

可以先得出一个粗糙的解，就是所有位置都单独覆盖，那么代价是$O(m)$的，而覆盖一个高度为h的子金字塔的代价是$O(h^2)$的，所以我们只需要考虑覆盖高度为$O(sqrtm)$范围内的子金字塔。

可以把金字塔的编排做一下修改如下图：



这样一个子金字塔可以用一个直角三角形来描述，设$f[i][j]$为已经做到第i列，当前的三角形高度为j的最小代价，转移时选择放更高的三角形或者把高处的进行单独覆盖，数组可以滚动使用。

时间复杂度$O(n\sqrt{m})$，空间复杂度$O(n+m)$。

13 Jeff and Removing Periods

【题目来源】

Codeforces 351D

【简要题意】

给定一个长度为n的数列A。

一个数列的“美丽度”的定义为执行以下操作消完这个数列的最小操作次数，一次操作分为三个步骤，如下：

（1）选择三个整数v, t, k，使得$a_v = a_{v+t} = a_{v+2t} = ... = a_{v+tk}$。

（2）删除a_v，a_{v+t}，$a_{v+2t}...a_{v+tk}$，对剩下的数重新标号。

（3）任意得重新排列这个数列。

有q组询问，每次询问数组$A[l..r]$的美丽度。

数据范围：$n, m, A_i \leq 10^5$

【解题思路】

因为可以重新排列，所以答案一定是颜色数或者颜色数+1。现在的问题就是要求出颜色数以及是否有一种颜色可以在不重排的前提下一次消光。

求颜色数是经典问题，可以把询问按右端点排序，把每种颜色放在尽可能靠右的位置即可，然后可以用同样的方法求出不能一次消完的颜色数，树状数组维护。

时间复杂度$O(q \log n)$，空间复杂度$O(n + q)$。

14 Doodle Jump

【题目来源】

Codeforces 346E

【简要题意】

有n个平台，分别在$ix \ mod \ p$的高度位置。

一种名为"$Doodler$"的生物每次最多可以跳h的高度，问在给定的这些平台上，它是否能从0开始，逐级跳过高度p。

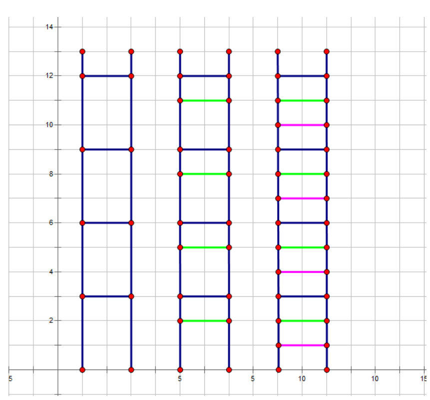
p是质数，T组询问。

数据范围：$T \leq 10000$，$n, h, p \leq 10^9$

【解题思路】

问题可以转化为求高度差最近的两个平台的高度差。

平台是在$mod \ p$域下进行分布的，所以可以把平台的构成分阶段来看待，如下图是一个在$mod \ 13$域下的例子。



首先，可以采取贪心策略，最优解一定可以取在最上端的那一段中，因为在$mod\ p$循环节下，每一段的构成的一样的，而最上端的那一段的平台数是最少的，而这会把我们的问题引入一个新模域x下，而新的x变为了$p\ mod\ x$，于是可以递归来解决子问题。

但是x转化到$p\ mod\ x$并不能给与复杂度的保证，因为当$x = p - 1$时，递归深度将会达到p。

而我们发现从下网上放阶梯和从上往下放阶梯是一样的，所以新的$p\ mod\ x$如果大于$\frac{p}{2}$，我们可以等价取用$p - (p\ mod\ x)$，来保证复杂度。

时间复杂度$O(T \log p)$，空间复杂度$O(1)$。

15 Pumping Stations

【题目来源】

Codeforces 343E

【简要题意】

给出一个n个点m条边的无向流网络，设$Flow(u,v)$为u到v的最大流，构造一个n的排列I，使得$\sum_{i=1}^{n-1} Flow(I_i, I_{i+1})$最大。

数据范围：$n \leq 200$，$m \leq 1000$

【解题思路】

我们需要用GomoryHu tree来构造解。

在GomoryHu tree上，两个点间的最大流即为两个点的树上路径上的最小值。构建这棵树我们可以用递归构建的方法，先随便选取两个点，然后做一次最大流，把最大流看作最小割把点集分成两个，以这次的最小割作为这两个点集之间的边，然后递归处理这两个点集，这样只需要做$n-1$次最大流运算就可以构出这棵树。

构出这棵树之后，现在的问题就是构造最大的解，由于最大流（最小割）是树上路径的最小值，于是我们肯定希望权值小的边的经过次数尽可能少。

可以设计出这样的构造方案：找到权值最小的边，为了使这条边只用一次，把这条边断了之后递归构造两个连通块之后，再把这两个连通块连起来。

这样可以使得每条边都被用到一次，因为GomoryHu tree的性质是取路径上的最小值，所以这样的构造是最优的。

时间复杂度$O(n^3m)$，空间复杂度$O(n+m)$。

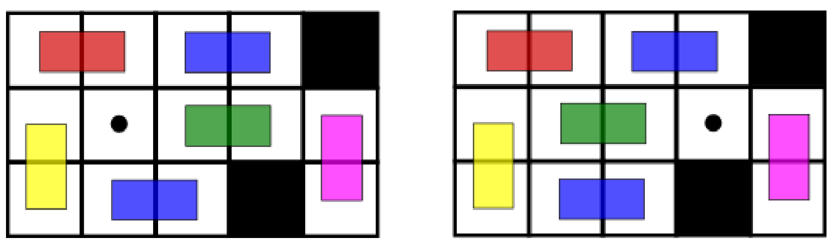
16 Xenia and Dominoes

【题目来源】

Codeforces 342D

【简要题意】

有一个$3 * n$的方格图，这些格子有些是障碍，有一个地方必须是空格子，剩下的格子放满多米诺骨牌，求有多少种放多米诺骨牌的方案，且骨牌必须能至少移动一次（如下图是一次移动），方案数mod 1000000007。



数据范围：$n \leq 10^4$

【解题思路】

骨牌要能移动，那么空格的四周一定存在方向确定的骨牌，可以强制把确定的骨牌用障碍填充，进行容斥，空格也相当于障碍。

剩下就是要计算放骨牌的方案数，状压DP。

时间复杂度$O(n)$，空间复杂度$O(n)$。

17 Candies Game

【题目来源】

Codeforces 341E

【简要题意】

有n个箱子，第i个箱子中有a_i个糖果。

我们可以执行这样的操作，选取两个箱子u，v，若$a_u > a_v$，那么在u中取a_v个糖果放到v中。

构造一种方法使得最后能把所有糖果放到恰好两个箱子里。

数据范围：$n \le 1000$，$a_i \le 10^6$

【解题思路】

考虑每次选取3个箱子，然后我们的目标是清空其中一只箱子。

设这3个箱子中的糖果数为A，B，C，且$A \le B \le C$。这时我们把$\lfloor \frac{B}{A} \rfloor$表示为二进制，然后从$B$ 和C 中取糖果给A，这个过程中会不断把A倍增，最后把B减到$B\ mod\ A$，而$C \le B$从而保证最后$C > 0$。这样一次操作后将会将最小数从A变成$B\ mod\ A$，可以构造出十分优秀的解。

这个过程类似于欧几里德辗转相除法，但是不能完美地每次使最小的数减少到$\frac{1}{2}$以下，但是在给定是数据范围内是可以良好运作的。

时间复杂度$O(n \log^2 a_i)$，空间复杂度$O(n)$。

18 Three Swaps

【题目来源】

Codeforces 339E

【简要题意】

有一个长度为n的数组A，$A_i = i$，现在这个数组被随机选取了3个区间翻转了，求出任意一组可行的区间（保证有解）。

数据范围：$n \leq 1000$

【解题思路】

一个区间翻转后会产生一些不和谐的位置，如$|A_i - A_{i-1}| \doteq 1$，然后枚举这些不和谐位置为区间两端搜索，由于不和谐位置是数量很少（只翻转了三次），所以可以在时限内找到解。

时间复杂度$O(n^2)$，空间复杂度$O(n)$。

19 GCD Table

【题目来源】

Codeforces 338D

【简要题意】

设二维数组$G[i][j] = gcd(i, j)$，对于给定的长度为k的数列a，求数列a是否在G中的某行连续出现。

数据范围：$n, m, a_i \leq 10^{12}$，$k \leq 10000$

【解题思路】

最早出现的行一定在$lcm(a_i)$，那么设列为y，于是可以列一系列方程$y + i \equiv 0\ (mod\ a_i)$。

解这个用中国剩余定理即可，由于模数不互质，所以需要分解模数来做。

最后还需要验证以下解是否是正确可行的。

时间复杂度$O(k \log n)$，空间复杂度$O(k)$。

20 Optimize!

【题目来源】

Codeforces 338E

【简要题意】

优化这份代码：

```
getAnswer(a[1..n], b[1..len], h)
  answer = 0
  for i = 1 to n-len+1
    answer = answer + f(a[i..i+len-1], b, h, 1)
  return answer

f(s[1..len], b[1..len], h, index)
  if index = len+1 then
    return 1
  for i = 1 to len
    if s[index] + b[i] >= h
      mem = b[i]
      b[i] = 0
      res = f(s, b, h, index + 1)
      b[i] = mem
      if res > 0
        return 1
  return 0
```

数据范围：$1 \le len \le n \le 150000$，$a_i, b_i, h \le 10^9$

【解题思路】

问题相当于求，有多少区间$[l,r]$，满足$r-l+1=len$，且把这段区间中的a_i降序排序后b_i升序排序，$a_i+b_i\geq h$。

考虑对每个a_i二分出最多与哪些标号的b_i能满足约束，设能和1到c_i的b满足约束，则是要使得$c_i-i\geq 0$，然后每次把区间向右移动一位，也就是删一个数，加一个数，对应到平衡树里就是对一个区间+1，-1，判断全局最小值是否大于等于0即可。

时间复杂度$O(n\log n)$，空间复杂度$O(n)$。

21 Rectangle And Square

【题目来源】

Codeforces 335D

【简要题意】

有n个平行于坐标轴的矩形，矩形的四角是整点(x_i, y_i)，求一个矩形的子集使得这些矩形的并是正方形。保证矩形不交。

数据范围：$n \leq 100000$，$x_i, y_i \leq 3000$

【解题思路】

可以看出正方形的四个顶点一定是某些矩形的顶点，可以枚举正方形的左上角顶点，再枚举边长进行判断。

这样做是$O(3000n)$的，但是由于边长必须满足另外顶点在矩形顶点上，所以可以取左上角顶点正下方和正右方点数较小的方向进行枚举，即直接枚举另一个顶点，这样做能构造出的均摊最大的数据是$O(n\sqrt{n})$的（一个$\sqrt{n} * \sqrt{n}$的点阵）。

可以这样判断正方形完全由矩形构成：正方形内部点全部被矩形覆盖，正方形的四条边上的点全为矩形对应边上的点。可以利用预处理前缀和来做到$O(3000^2) - O(1)$判断。

时间复杂度$O(n\sqrt{n} + 3000^2)$，空间复杂度$O(n + 3000^2)$。

22 Buy One，Get One Free

【题目来源】

Codeforces 335F

【简要题意】

n个物品，获得第i个物品的代价为A_i，付出代价获得物品i后可以选择一个物品j(满足$A_j < A_i$)直接获得。求获得所有物品的最小代价。

数据范围：$n \le 500000$，$A_i \le 10^9$

【解题思路】

首先可以设计一个$O(n^2)$的DP，设$f[i][j]$表示转移到第i种价格，还可以免费拿j个物品时的最大省钱量，那么每次可以枚举当前价格的物品免费获得的数量进行转移。把DP的状态记录略微修改下，设$f[i][j]$为转移到第i种价格，已经免费拿了j个的最大省钱量，那么$f[i][j]$与$f[i][j-1]$ 的差值将会有特殊的含义。$f[i][j] - f[i][j-1]$的值表示多配对一对物品能增加的获利，增加的获利的来源可能是直接增加一组配对，或拆除原先的几组配对，加入两个新物品来组成新的一系列配对。也就是说DP值的增量对应着一些具体的配对操作。

那么我们就可以用这些增量来描述这个DP阶段，由于增量对应着一些具体的操作，所以这些增量是递减分布的(一些有关联的增量，即修改再修改再修改等情况产生的增量是递减的，证明见后文，可以先认为这个成立)，不然对于j 较小的$f[i][j]$就不优了。我们可以采取一定的贪心策略来维护这些增量来取代暴力的DP。将物品按价格排序后，从大到小处理，对于新加进来的一些相同价格物品我们可以采取下面两种措施：

(1)和还没有配对的老物品配对。

(2)撤销一些已有的配对，用新空出来的物品来配对新物品。

设新物品的价格为A，已有配对造成的增量为P。假如我们采取(1)操作，我们的获利为A，而采取(2)操作时我们的获利为$2A-P$，似乎看不出哪种更优，但我们可以进一步分析这些操作对答案的贡献。

由于我们的算法是基于用贪心维护DP数组的，所以我们维护的增量必须要保证维护到最大配对数，这样我们的DP转移才完整，而(1)操作只需要1个新物品就可以增加一组配对，而(2)操作则需要2个新物品才能增加一组配对，所以我们应优先进行(1)操作。

对于(2)操作，我们显然会选择最小的P来修改，查找最小的P我们可以用堆维护增量来完成。对于P值的不同，我们又会遇到两种情况：

(1)当$P < A$时，我们需要把原先的P替换成A，并加入一个A。这一步的含义是，我们根本不进行P这个操作了，而把P操作加入的2个元素(这两个元素必定都大于A)分别与A配对，获得的两个增量都为A。

(2)当$P \geq A$时，我们需要保留原先的P(因为P在配对数较小时比A优)，再加入一个$2A-P$，这里可以看出$2A-P \leq P$，所以也就顺带着证明了我们前文留下的问题——有关联的操作对答案的贡献也是递减的，所以我们可以直接用堆维护增量。

算法看上去就这样结束了，但是还有问题。我们进行操作(2)时，每次需要消耗2个当前物品，操作完成后当前物品可能还剩1个，这个时候我们还可以进行一种操作，就是取出一个$P(P < A)$，把它替换成A，这种操作的含义是撤销这一个操作，把余出的2个老物品的其中一个和剩下的新物品进行配对。

时间复杂度$O(n\log n)$，空间复杂度$O(n)$。

23 Lucky Tickets

【题目来源】

Codeforces 333C

【简要题意】

k $lucky$序列的定义为把这个数字序列每一位拆开，可以在一些数的左边或右边加上‘+’‘-’‘*’或‘(’‘)’使得最后答案等于k。现在给出k，求出m个不同的长度为8的数字序列。

数据范围：$0 \le k \le 10^4$，$1 \le m \le 3 * 10^5$

【解题思路】

把8位分为4位和4位，枚举4位和符号，用剩下4位去拟合k。

前4位和后4位可以交换，所以用这种算法可以构造出足够的解。

时间复杂度$O(10^4)$，空间复杂度$O(10^4)$。

24 Theft of Blueprints

【题目来源】

Codeforces 332D

【简要题意】

给出一个n个点的带权无向图，这张图满足一个性质：对于任意一个大小为K的点集合S，恰好有一个点与S中每一个点都有边。令这个点为v_S，并且对S进行操作的代价是S中每个点与v_S的边权之和。

求对于一个大小为K的子集操作代价的期望，即$e(\sum_{u|u\in S} value(u, v_S))$，输出整数部分。

数据范围：$K \le n \le 2000$

【解题思路】

问题可以转化为求

$$\frac{\sum_{S,|S|=K} \sum_{u|u\in S} value(u, v_S)}{C_n^K}$$

设d_i为i号点的度数，那么

$$\sum_{S,|S|=K} \sum_{u|u\in S} value(u, v_S)$$

$$= \sum_{u=1}^{n} \sum_{v=u+1}^{n} value(u, v) * (C_{d_u-1}^{K-1} + C_{d_v-1}^{K-1})$$

由于精度要求很低，所以可以直接用$double$类型暴力计算。

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

25 Binary Key

【题目来源】

Codeforces 332E

【简要题意】

构造一个长度为k的01串q，对于给定的串q，通过下面这份代码运行后生成串s。

```
i = 0;
j = 0;
s = <>;
while i is less than the length of the string p
{
    if q[j] == 1, then add to the right of string s character p[i];
    increase variables i, j by one;
    if the value of the variable j equals the length of the string q, then j = 0;
}
```

求最小字典序的串q。

数据范围：$k \le 2000$，$|s| \le 200$，$|p| \le 10^6$

【解题思路】

为了使字典序最小，可以枚举串q中1的个数贪心，尽可能把1放在较后的位置，同时1的个数也可以确定该字符在串p中的位置。

算出所有可能答案后取字典序最小的。

时间复杂度$O(|s|^2k)$，空间复杂度$O(|s|+|q|+k)$。

26 The Great Julya Calendar

【题目来源】

Codeforces 331C

【简要题意】

给出一个数n，每次可以把n减去n十进制表示下的某一位上的数字，求最少多少次可以把n减到0。

数据范围：$n \leq 10^{18}$

【解题思路】

贪心，每次减去各位上最大的数字是最优的。

DP预处理后i位是9999999j格式，高位上最大的数字是k的把9消完的最小次数，这样可以快速算出n的步数，而DP计算可以直接递归解决子问题。

时间复杂度$O(10^3 \log n)$，空间复杂度$O(10^2 \log n)$。

27 The Evil Temple and the Moving Rocks

【题目来源】

Codeforces 329D

【简要题意】

在一个$n*n$的方格图上有四种（上、下、左、右）的石块，当一个石块被撞击时就会往那个方向移动，当撞到其他石块或边界时就会停止。构造一个这样的方格图并选取一个初始移动石块，时撞击发生至少n^3-n^2次。

数据范围：$n\le 300$，n是偶数

【解题思路】

用以下格式的方块铺满全图即可：

>>>>>>>>>> . > . > . > . > . > .v

ˆ. < . < . < . < . < . <<<<<<<<<<

时间复杂度$O(n^2)$，空间复杂度$O(1)$。

28 Reclamation

【题目来源】

Codeforces 325D

【简要题意】

有一个$r * c$的地图，把左边界和右边界粘起来使得形成一个圆柱，现在要不断地挖去其中的m个格子，要求任何时候都存在一条从最上方到最下方的路径（四联通），如果某次操作不满足要求则不做，问最后有多少次操作是成功的。

数据范围：$r, c \leq 3000$，$m \leq 3 * 10^5$

【解题思路】

可以把地图复制一边放到右边，地图是循环的。

没有从上到下路径的情况就是有一条从左到右的横断路径，用并查集维护同一个方格的左右的联通情况判断。

时间复杂度$O(rc + m)$，空间复杂度$O(rc)$。

29 The Red Button

【题目来源】

Codeforces 325E

【简要题意】

构造一个长度为$n+1$的序列，设这个序列为A，使得1到$n-1$各出现一次，0在首尾各出现一次，且A_{i+1}为$2*A_i\ mod\ n$或$(2*A_i+1) mod\ n$。

数据范围：$n \le 10^5$

【解题思路】

首先把x和$x+1$（x为偶数）并成一组，那么每个点的出边指向为一个组，而且x和$x+1$指向的组是不同的。

那么就可以在新的图上求出一个欧拉回路，然后根据出边选择使用组中的哪一个元素即可。

求欧拉回路是经典问题，只需要对图进行DFS遍历，退栈时加入答案序列中就可以求出倒序的欧拉回路。

时间复杂度$O(n)$，空间复杂度$O(n)$。

30 Tournament-graph

【题目来源】

Codeforces 323B

【简要题意】

构造一张n个点的有向完全图，使得任意最短路$dist(u,v) \leq 2$，或判断无解。

数据范围：$3 \leq n \leq 1000$

【解题思路】

$n = 4$的时候很特殊，无解。

可以先构造出一个环$1-2-3-4-...-n-1$，然后对于一对$i, j(i+2 \leq j)$，如果$i \equiv j(mod\, 2)$则i向j连边，否则j向i连边。

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

31 Two permutations

【题目来源】

Codeforces 323C

【简要题意】

给出一个长度为n的排列p，一个长度为n的排列q，m次询问，询问p中区间$[lp_i, rp_i]$中有多少数在q区间$[lq_i, rq_i]$中。强制在线。

数据范围：$n \le 10^6$，$m \le 2 * 10^5$

【解题思路】

由于p和q都是排列，所以可以把p中的数都映射为在q中的位置，这样问题就转化为求区间内某个范围内的数的个数。

把p中的新数按大小排序逐一加入函数式线段树维护即可。

时间复杂度$O((n+m)\log n)$，空间复杂度$O(n\log n)$。

32 Ciel and Flipboard

【题目来源】

Codeforces 321D

【简要题意】

有一个n行n列的板子，每个格子上有一个数字。大家都知道n是一个奇数，不妨设$x = \frac{n+1}{2}$。可以选择一个x行x列的子矩阵，并将其中的所有元素乘-1。他可以使用这个操作任意多次。目标是最大化板子上的数字和，求最大数字和。

数据范围：$n \le 33$，n是奇数

【解题思路】

设$x = \frac{n+1}{2}$，$f[i][j]$表示这一格是否乘了-1，由观察易得：

$f[i][j]\ xor\ f[i][x]\ xor\ f[i][j+x] = 0\ (j \le x)$

$f[i][j]\ xor\ f[x][j]\ xor\ f[i+x][j] = 0\ (i \le x)$

那么可以枚举$f[i][x]$这一列的x个，然后剩下的四个一组互不相关，可以直接贪心。

时间复杂度$O(2^{\frac{n+1}{2}} * n^2)$，空间复杂度$O(n^2)$。

33 Have You Ever Heard About the Word?

【题目来源】

Codeforces 319D

【简要题意】

给出一个长度为n的字符串S，每次把最短的最靠左的连续的重复串，删除其中一个，求这个串最后的形态。

数据范围：$n \leq 50000$

【解题思路】

每次被删除的串的串长一定是不降的，所以可以先考虑如何判断是否存在某种长度的连续重复串。

假设要判断是否存在长度为L的连续重复串，那么可以把串右移L位，逐位比较，找出连续的长度为L的匹配位进行删除。这个的朴素实现是总时效$O(n^2)$的，但是可以这样优化，枚举位置xL，Hash判断LCP和LCS，如果大于等于L则存在连续重复串。这一步的复杂度为$O(n \ln n \log n)$（调和级数）。

然后就暴力把连续重复串去掉，由于串长的限制，所以最多只会去掉$O(sqrtn)$种，所以这一步的复杂度为$O(n\sqrt{n})$。

时间复杂度$O(n \ln n \log n + n\sqrt{n})$，空间复杂度$O(n)$。

34 PE lesson

【题目来源】

Codeforces 316D

【简要题意】

有n个人在传球，有些人最多可以传一次，另一些人最多可以传两次，一开始每个人手上球的编号是自己的编号，求完成传球后所有人手上的求的编号的排列种数$mod\ 1000000007$。

数据范围：$n \le 10^6$

【解题思路】

可以发现最后的排列可以看作一些置换圈，而构造这些置换圈的最优方法只可以在一个圈中容纳最多两个传一次的人。

设A个人可以传一个，B个人可以传两次。

考虑先把这A个人完成配对，把其中一些人配成一对（在同一个置换圈内），那么可以设计一个递推式$f_i = (i-1)f_{i-2} + f_{i-1}$，表示这个人选一个人配对$(i-1)f_{i-2}$或不配对$f_{i-1}$。

然后把那B个人插入这些置换圈中，可以设计递推式$g_i = (A+i)g_{i-1}$，有$A+i-1$个位置可以插，或者单独放。

那么最终答案为$f_A * g_B$。

时间复杂度$O(n)$，空间复杂度$O(n)$。

35 Summer Homework

【题目来源】

Codeforces 316E

【简要题意】

对于一个长度为n的整数序列A，需要支持以下操作，操作数为m:

（1）将A_u修改为v。

（2）将A_l～A_r加上$delta$。

（3）求$\sum_{i=0}^{r-l} A_{l+i} * fib_i \ mod \ 1000000000$其中$fib_0 = fib_1 = 1, fib_i = fib_{i-1} + fib_{i-2}$。

数据范围：$n, m \leq 200000$

【解题思路】

对于询问中的斐波那契数列，可以用线段树上维护一个矩阵来完成，对于修改（1）只需要修改线段树叶子节点的矩阵，然后到根路径更新即可，对于修改（2）只需要预处理矩阵的前缀和打标记即可。

时间复杂度$O(m \log n)$，空间复杂度$O(n)$。

36 Good Substrings

【题目来源】

Codeforces 316G

【简要题意】

给出一个字符串S，以及n个约束，每个约束为该串在串s_i中的出现次数在$[l_i, r_i]$范围内，求S有多少形式不同的子串满足这些所有约束。

数据范围：$|S| \leq 50000$，$n \leq 10$

【解题思路】

对S串建立后缀数组，对于每个后缀可以二分出满足所有约束的长度，然后求本质不同的子串是经典问题，利用后缀数组的$height$数组可以解决。

时间复杂度$O(|S| \log^2 |S|)$，空间复杂度$O(|S| \log |S|)$。

37 Sereja and Squares

【题目来源】

Codeforces 314E

【简要题意】

起初有一个字符串S，由大写字符、小写字符构成，小写字符和大写字符是配对的，即左边的一个小写字符和右边的一个对应大写字符配对，且这些配对形成一个括号序。

现在这个串中的所有大写字符和一些小写字符被隐藏了，变成了问号，现在给出新的串S，求原串有多少种可能mod 4294967296。

数据范围：$|S| \le 10^5$

【解题思路】

这题并没有任何复杂度较优的解法，官方题解为卡常数。

设$f[i][j]$为完成了前i个，配对了j对的方案数，数组可以滚动。

时间复杂度$O(|S|^2)$，空间复杂度$O(|S|)$。

38 Fetch the Treasure

【题目来源】

Codeforces 311C

【简要题意】

有h个位置编号1到h，一开始在1号位置，现在有一些位移参数a_i，表示能够走到编号为$1 + \sum v_i * a_i$ $(v_i \ge 0)$ 的位置，一开始只有一个参数k。这h个位置中，有n个位置上有权值，描述为位置x_i上有权值c_i。

需要支持以下操作，共m次：

（1）增加一个位移参数。（这种操作不超过20个）

（2）把c_i修改为$c_i - value$

（3）找到能走到的位置中权值最大的那个，如果有多个权值最大，就选择读入时最早的，输出那个权值，并把那个位置的权值赋为0。

数据范围：$x_i \le h \le 10^{18}$，$n, m \le 10^5$，$k \le 10^4$，$c_i \le 10^9$

【解题思路】

由于位移参数的修改次数很少，所以可以考虑每次修改时重新计算能到达的位置。

设现在的位移参数为集合A，由于一开始的第一个参数$k \le 10^4$，所以可以用最短路算法（SPFA、Dijkstra）计算出$dist_i$ 表示走到$mod\ k = i$的位置最小距离，即能走到的第一个$mod\ k = i$的位置，那么如果$x_i \le dist_{x_i\ mod\ k}$，$x_i$这个位置就可以走到。

对于操作（2）（3），只需要维护一个带映射表的堆，或者直接开一棵线段树来进行计算。

时间复杂度$O(20k\log k+m\log n)$，空间复杂度$O(n+k)$。

39 Biologist

【题目来源】

Codeforces 311E

【简要题意】

给出一个长度为n的01序列和m条要求（给出一个集合，集合大小为k_i，要求这个集合中的数都是0或要求都是1），满足一些要求会得到钱，不满足一些要求会失去钱，可以花费一些钱来修改一个数的01状态，问最大获利（负表示亏损）。

数据范围：$n \le 10^4$，$m \le 2000$，$k_i \le 10$

【解题思路】

可以建立最小割模型来解决此题，先把所有正权都加到答案中，这样就是求最小支出。

对于每个数，根据它的01状态向源汇连边，流量是修改它的花费。对于每个要求，给这个要求建一个点，根据要求集合中是0还是1 来确定是要求点和集合点之间的连边。这样就可以选择割集合点（改变01状态）和割要求点（不满足要求付出代价），用最大流（最小割）解决。

这样建图的点规模和边规模都是$O(n+m)$的。

时间复杂度$O((n+m)^3)$，空间复杂度$O(n+m)$。

40 Context Advertising

【题目来源】

Codeforces 309B

【简要题意】

一个文本由一些用空格隔开的单词构成，单词个数为n，总长度为L，现在规定每行只能容纳r个字符（包括空格），可以写c行。要从原文本中选择连续的一段，使得连续分配到几行后，单词数最多。输出方案。

数据范围：$1 \leq n, r, c \leq 10^6$，$r * c \leq 10^6$，$L \leq 5 * 10^6$

【解题思路】

可以用倍增算出以某个单词为末尾的最长段落，方案也可以直接输出。

时间复杂度$O(L + n \log c)$，空间复杂度$O(L)$。

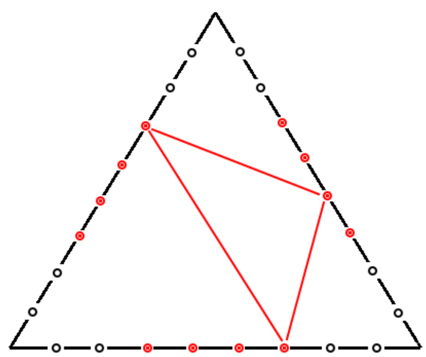
41 Tennis Rackets

【题目来源】

Codeforces 309D

【简要题意】

有一个边长为n的正三角形，在每条边上选一个离最近顶点距离不超过m的整点，使这三个点连成的三角形为钝角三角形，求点对数。



数据范围：$n \leq 32000$，$0 \leq m \leq \frac{n}{2}$

【解题思路】

枚举其中两条边上的点的位置，另一条边上的可行点用单调性计算，可以构造一个坐标系用点积来判断钝角，一些冗余的计算可以直接乘系数而不枚举。

时间复杂度$O(n^2)$，空间复杂度$O(1)$。

42 White, Black and White Again

【题目来源】

Codeforces 306C

【简要题意】

有w件两两不同的好事和b件两两不同的坏事，每天要么全部发生好事，要么全部发生坏事。共有n天，先若干天发生好事，再若干天发生坏事，再若干天发生好事。每天不会不发生事，求有多少种发生事的方案$mod\ 1000000009$。

数据范围：$n, w, b \le 4000$

【解题思路】

预处理组合数，枚举先发生坏事的天数，剩下是组合数计算。

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

43 Polygon

【题目来源】

Codeforces 306D

【简要题意】

构造一个n个点，n个角都相同，n条边长度都不同的多边形。

边长要求在$[1,1000]$范围内，精度为10^{-3}。

数据范围：$n \le 100$

【解题思路】

首先$n < 5$时无解。

由于每个角要求相同，所以可以先算出角度，然后每次按那个角度偏转，把第一个点放在原点，然后第i条边长度为$100 + 0.02 * i$，最后一个点位置直接计算出来即可。

时间复杂度$O(n)$，空间复杂度$O(1)$。

44 Olya and Graph

【题目来源】

Codeforces 305D

【简要题意】

给出一张n个点m条边的有向图，求有多少种加边方案能使图满足以下性质：

（1）所有边(u, v)满足$u < v$，且没有重边。

（2）对于所有点对(u, v) $(u < v)$满足，u到v的最短路为$v - u$或$v - u - K$。

答案mod 1000000007。

数据范围：$n \le 10^6$，$m \le 10^5$

【解题思路】

考虑该图性质，易得边只有两种$(u, u+1)$和$(u, u+K+1)$，且第一种边必须全满，不然对于$u + K \le v$的点对就无法满足性质，对于第二种边，不能出现点对(u, v)间可以走超过1次这类边，不然也无法满足约束。

所以只需要考虑第二种边的情况，把约束写得更清楚一点就是第二种边第一次出现和最后一次出现必须相交，即不能存在$(u, u+K+1)$和$(v, v+K+1)$使$u + K + 1 \le v$，可以枚举最后一次出现的位置计算。

时间复杂度$O(n + m)$，空间复杂度$O(n + m)$。

45 Playing with String

【题目来源】

Codeforces 305E

【简要题意】

两个人在玩一个博弈游戏，一开始有一个串T，每次可以选择一个位置i，满足$T_{i-1} = T_{i+1}$，然后从这个位置把串撕开，变成两个串$T[1..i-1]$，$T[i+1..|T|]$，最后不能操作的人为输。

求先手是否有必胜策略，如果有输出最小的可行的第一次操作的位置。

数据范围：$|T| \le 5000$

【解题思路】

可以把这个串变成一个01串，1表示这个地方可以断开，然后若干的几段1是互不相干的，然后就可以计算连续一段1的SG值，来得出先手是否必胜。

求最小操作只需要枚举位置用SG值判断即可。

时间复杂度$O(|T|^2)$，空间复杂度$O(|T|)$。

46 Random Ranking

【题目来源】

Codeforces 303E

【简要题意】

有n个考生，第i个考试的分数等概率分布在$[l_i, r_i]$，求第i个人考第j名的概率。

数据范围：$n \le 80$，$1 \le l_i < r_i \le 10^9$

【解题思路】

可以把时间离散分段，现在考虑第i个人分数在某一段中考第j名的概率。

设第i个人考了x分，那么考第j名的概率就是

$$f(x) = \sum_{|S|=j-1} (\prod_{i \in S}(x - l_i))(\prod_{i \notin S}(r_i - x))$$

那么我们就是要求

$$\int_l^r f(x)dx$$

$f(x)$的多项式可以用DP得出，但是这个直接用定积分算会炸精度，所以需要考虑其他算法。

其实可以直接计算$f[i][j]$为i个人$< l$，j个人$\in [l, r]$的概率，当$j+1$个人在同一区间中等概率分布时，得到任何名次的概率都是$\frac{1}{j+1}$。

剩下的就是简单的DP计算。

时间复杂度$O(n^5)$，空间复杂度$O(n^2)$。

47 Yaroslav and Algorithm

【题目来源】

Codeforces 301C

【简要题意】

构造一个算法，使得该算法能把所有输入的数n加一输出。算法的定义如下：

（1）这个算法接受一个字符串作为输入。我们设这个输入字符串为a。

（2）这个算法由一些命令组成。i号命令的形式为"$s_i >> w_i$"或"$s_i <> w_i$"，其中s_i和w_i是长度不超过7的字符串（可以为空），由数字或字符“?”组成。

（3）这个算法每次寻找一个编号最小的命令i，使得s_i是a的子串。如果没有找到这样的命令，那么整个算法终止。

（4）设找到的命令编号为k。在字符串a中，s_k第一次出现的位置会被w_k替换。如果这个命令形如"$s[k] >> w[k]$"，那么这个算法继续执行（回到第3步）。否则，算法终止。

（5）算法的输出就是算法终止时字符串a的值。

要求命令条数不超过50，执行次数不超过200。

数据范围：$n \leq 10^25$

【解题思路】

最主要的问题是解决进位。构造出的算法流程如下：

（1）在串头放一个问号。

（2）把问号移到串末。

（3）把问号变成双问号。

（4）不断把双问号前的9变成0，然后把双问号往前移，直到不是9。

该算法的命令条数为22。

时间复杂度$O(1)$，空间复杂度$O(1)$。

48 Yaroslav and Arrangements

【题目来源】

Codeforces 301E

【简要题意】

称一个长度为n的数列a是好的，需要满足以下条件：

（1）$|a_i - a_{i-1}| = 1$，$|a_1 - a_n| = 1$

（2）$a_1 = min_{i=1}^{n} a_i$

称一个长度为n的数列b是优秀的，需要满足以下条件：

（1）数列中的元素不降

（2）数列中的所有数$1 \le b_i \le m$，$b_i \in Z$

（3）通过将b重排后可以得到至多k个不同的好数列。

对于给出的n, m, k，求不同的长度不超过n的优秀数列的数量$mod\, 1000000007$。

数据范围：$n, m, k \le 100$

【解题思路】

设$f[i][j][k][l]$为当前长度为i，用到第j种数字，空位有k个，重排方案为l，每次可以把新的一种数字放在前一种数字留的空位中来转移重排方案。DP数组可以滚动使用。

时间复杂度$O(\lfloor \frac{n}{2} \rfloor^4 k)$，空间复杂度$O(\lfloor \frac{n}{2} \rfloor^3 k)$。

49 Greg and Caves

【题目来源】

Codeforces 295D

【简要题意】

在一个$n*m$的白色方格图上，涂一个黑色连通块，使得这个连通块满足以下性质：

（1）黑色部分在每行中都是连续的，设第i行的黑色范围为$[l_i, r_i]$。

（2）找到一行mid，使得$|r_mid - l_mid|$是最大的（最大不一定要唯一）。

（3）对于$i < j \le mid$，满足$l_j \le l_i \le r_i \le r_j$。

（4）对于$mid \le j < i$，满足$l_j \le l_i \le r_i \le r_j$。

以上的i, j不计算未涂色的部分，求染色的方案数$mod\ 1000000007$。

数据范围：$n, m \le 2000$

【解题思路】

考虑把mid上下的两部分分开做，设$f[i][j]$为已经做了i层，当前层长度为j，那么$f[i][j]$对$f[i+1][k]$的贡献为$f[i][j] * (j - k + 1)$。做完之后只需要把上下两部分拼起来，为了避免重复，我们可以强制下部分第2层严格短于第1层。

这些DP转移的朴素实现是$O(nm^2)$的，但是可以利用计算前缀和优化到$O(nm)$。

时间复杂度$O(nm)$，空间复杂度$O(nm)$。

50 Shaass and Painter Robot

【题目来源】

Codeforces 294D

【简要题意】

在一张$n * m$的白色方格图上染色，一开始在(S_x, S_y)，有一个初始方向，开始染色，走到边界就根据镜面反射原理进行反射，直到整张方格图黑白相间时才停下，问总共走了多少距离，或不可能停下。

数据范围：$n, m \le 10^5$

【解题思路】

可以按走到边界上的位置来划分时间，当边界上的所有相隔的位置都被走到了，那么整张方格图就黑白相间了，所以只需要模拟$O(n+m)$次反射就可完成计算和判断，并记录路程。

时间复杂度$O(n+m)$，空间复杂度$O(n+m)$。

51 Distinct Paths

【题目来源】

Codeforces 293B

【简要题意】

一个$n*m$的方格图，一些格子已经被涂上给出的k种颜色中的一种。你需要把每个没涂色的格子涂色使得从左上角到右下角的每条路径都不会经过两个颜色一样的格子。路径只能向右或向下走。输出答案mod 1000000007。

数据范围：$n,m \le 1000$，$k \le 10$

【解题思路】

如果$n+m-1>k$那么是一定不可行的，所以只需要考虑n,m较小的情况。

由于n,m都较小，所以考虑搜索来解决，而直接搜的效率是$O(nmk^{n*m})$的，当$n+m-1=k$时，由基本不等式可得$n*m \le (\frac{k+1}{2})^2$，所以复杂度为$O(nmk^{(\frac{k+1}{2})^2})$，还需要优化。

由于颜色的顺序并无关系，所以考虑把染色最小表示，经过测试后最小表示后的状态在10^6种以内，于是就有潜在的保证，那么在搜索枚举的时候值需要按最小表示的状态枚举，最后用组合数计算每种最小表示的原染色方案数。

时间复杂度$O(nmk^{(\frac{k+1}{2})^2})$，空间复杂度$O(nm)$。

52 Ksusha and Square

【题目来源】

Codeforces 293D

【简要题意】

给出一个凸多边形，共n个顶点(x_i, y_i)，在这个凸多边形中随机选两个整点，求以这两个整点连线为对角线的正方形的期望面积。

数据范围：$n \leq 10^5$，$|x_i|, |y_i| \leq 10^6$

【解题思路】

设选取的两个点为(x_1, y_1)和(x_2, y_2)，欧几里德距离为d，那么正方形的面积即为$\frac{d^2}{2}$，而$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$，所以横纵坐标可以分开考虑。

由于是凸多边形，所以整个多边形只会有完整的一圈，所以可以暴力统计每条线段下方有多少点，然后利用前缀和计算。

时间复杂度$O(n + |x|)$，空间复杂度$O(n + |x|)$。

53 Close Vertices

【题目来源】

Codeforces 293E

【简要题意】

一棵n个点的树上，每条边有一个非负整数边权。定义路径长度为经过的边数，路径权重为路径上的边权和。

求有多少点对(u, v)满足路径长度小于L且路径权值小于W。

数据范围：$L \le n \le 10^5$，$W \le 10^9$

【解题思路】

点分治，对每个重心可以把路径按权值排序，树状数组维护长度计算。

时间复杂度$O(n \log^2 n)$，空间复杂度$O(n)$。

54　Polo the Penguin and Lucky Numbers

【题目来源】

Codeforces 288E

【简要题意】

定义幸运数为由4和7组成的数，设A_i为第i个幸运数（从小到大）。

给出两个幸运数l, r，假设$[l, r]$范围内有n个幸运数，第i个为a_i，求

$$\sum_{i=1}^{n-1} a_i * a_{i+1} \ \ mod\, 1000000007$$

数据范围：$l \le r \le 10^{100000}$

【解题思路】

可以用数位DP来解决这个问题。f_i为幸运数长度为i的幸运数数列的答案，那么只要在每个数前面加上4和7就可以计算f_{i+1}了，即

$$f_{i+1} = \sum_{i=1}^{2^i-1}(4*10^i + a_k)(4*10^i + a_{k+1})$$
$$+47777...777 * 7444...444$$
$$+\sum_{i=1}^{2^i-1}(7*10^i + a_k)(7*10^i + a_{k+1})$$

用数位DP的方法进行类似的运算即可。

时间复杂度$O(\lg l + \lg r)$，空间复杂度$O(\lg l + \lg r)$。

55 Tourists

【题目来源】

Codeforces 286D

【简要题意】

在一条数轴上有n堵墙，第i堵墙会在t_i时刻出现在$[l_i, r_i]$。

有m对人从原点开始以相同速度往右走，第i对人在时刻q_i出发，求每队人有多少实现不能看见对方。

数据范围：$n, m \le 10^5$，$l_i, r_i, t_i, q_i \le 10^9$

【解题思路】

每堵墙遮挡视线的时间是一个分段函数，其中时间较早和较晚时的贡献都是常数，中间一段是斜率为1的一次函数，可以用线段树预处理这n堵墙的所有函数的并集，然后直接在线段树上询问。

时间复杂度$O((n+m)\log t)$，空间复杂度$O(n \log t)$。

56 Ladies' Shop

【题目来源】

Codeforces 286E

【简要题意】

有一个正整数集合S，如果对于所有$u \in S, v \in S, u+v \leq m$则$u+v \in S$，那么称这个集合是闭合的。

首先先判断这个集合是否是闭合的，然后构造一个最小的正整数集合T，对集合T不停地做这样的操作：假如$u \in T, v \in T, u+v \leq m$那么把$u+v$加入集合$T$，最后需要$T = S$。

数据范围：$n, m \leq 10^6$

【解题思路】

判断集合S是否闭合，只需要做一次FFT卷积就可以了，判断是否还有$u+v$不在集合里。

对于构造集合T，也可以直接通过这次FFT的结果来完成，假如x这个数可以被$u+v=x$得到，那么不管u和v是否真的在集合T中，x的存在都是不必要的，因为一定会有一些T中的元素可以得到x。这样就可以确定集合T包含哪些元素，其实最优方案是唯一的。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

57 Positions in Permutations

【题目来源】

Codeforces 285E

【简要题意】

一个n的排列A，若$|A_i - i| = 1$则称第i位是完美的，求完美位置数为k的排列数$mod\ 1000000007$。

数据范围：$k \le n \le 1000$

【解题思路】

设$f(m)$为完美位置正好为m的排列数，$F(m)$为完美位置至少为m的方案数。

那么$F(m) = \sum_{i=m}^{n} C_i^m f(i)$，所以$f(m) = F(m) - \sum_{i=m+1}^{n} C_i^m f(i)$，这样如果我们已经计算完成了$F(m)$，我们就可以$O(n^2)$计算$f(m)$。

那么现在的问题就是计算$F(m)$，没有了“正好”这个限制，计算会变得简单很多，设$dp[i][j][2][2]$为已经做到第i位，已经完成了j个完美位置，以及i和$i+1$是否被使用，这个DP的复杂度为$O(n^2)$。

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

58 Cow Tennis Tournament

【题目来源】

Codeforces 283E

【简要题意】

有n个人，每个人由一个互不相同的能力值s_i描述，每两个人之间发生了一次比赛，能力值较大的人获胜。

现在强制改变了m组人的所有比赛结果，每组人的能力值是一个区间$[l_i, r_i]$。

求有多少三元组(p, q, r)满足p赢了q，q赢了r，r赢了p。（三元组可以当作集合看待）

数据范围：$n, m \leq 10^5$，$s_i \leq 10^9$

【解题思路】

考虑求出不满足条件的三元组，即有多少三元组仍然保持一开始的状态。

枚举三元组中能力值第二的人，求出能力值比他低被它打败的以及能力值比他高把他打败的人即可，可以用线段树来求解。

时间复杂度$O((n+m)\log n)$，空间复杂度$O(n)$。

59 k-Maximum Subsequence Sum

【题目来源】

Codeforces 280D

【简要题意】

对于一个长度为n的数列a，需要进行以下操作m次：

（1）把a_i赋值成x。

（2）询问区间$[l, r]$的不超过k段的最大不相交子段和。

数据范围：$n, m \le 10^5$，$a_i, x \le 500$，$k \le 20$

【解题思路】

不超过k段的最大不相交子段和可以用费用流来解决，完成连边后做出流量不超过k的最大费用流即可。

考虑优化这个算法，由于费用流的图其实是一个数列，而反边的流量其实是原流量的相反数，所以每次只需要求出这段区间的最大子段和，然后把这个子段的每个数乘-1，往复做k次或最大子段和为负时停止即可。

以上操作都可以用线段树直接维护。

时间复杂度$O(mk \log n)$，空间复杂度$O(n)$。

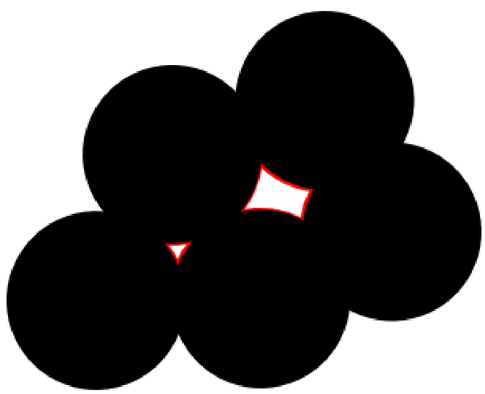
60 The Last Hole!

【题目来源】

Codeforces 274C

【简要题意】

在平面上放了n个圆，第i个圆的圆心在(xi, yi)。最开始所有圆半径都为0，然后所有圆同时开始变大，在时刻$t(t > 0)$所有圆的半径都为t。想象一些黑色的实心圆放在一个无穷大的白色平面上，每个时刻都会存在一些黑色和白色的联通快。需要注意一点，随着圆的增大，越来越多的圆会相交。



定义一个白色的封闭区域为一个“洞”，例如上图中包含两个红色边框的“洞”，随着圆的增大，一些新的洞会出现，也会有一些旧的洞消失。找一个最早的时刻，使得之后再也没有洞。

数据范围：$n \leq 100$，$|x_i|, |y_i| \leq 10^4$

【解题思路】

只有两种情况可能会产生“洞”：三个点为锐角三角形，四个点为矩形。

三角形的情况可以直接枚举三个点判断，矩形的情况也可以直接枚举，由于矩形的数量是$O(n^3)$级别的，所以也可以直接判断。

判断时需要列方程找出洞的最后一个点，判断这个点是否在洞消失之前就被覆盖了。

时间复杂度$O(n^4)$，空间复杂度$O(n)$。

61 Dima and Figure

【题目来源】

Codeforces 273D

【简要题意】

在一张$n * m$的白色方格纸上画黑色的连通块，使得连通块内任意两个格子的四连通最短路为曼哈顿距离。

计算画黑色连通块的方案数mod 1000000007。

数据范围：$n, m \leq 150$

【解题思路】

容易发现黑色连通块会是一个类似“凸四边形”的形状，左右界分别先扩大后缩小。

于是我们可以设计DP，$f[i][j][k][l][r]$表示前第i行，现在的左右界为$[j, k]$，左右界的扩大缩小情况是l和r的方案数，对于一个i可以设计$O(m^2)$的区间DP转移。

时间复杂度$O(nm^2)$，空间复杂度$O(m^2)$。

62 Maximum Waterfall

【题目来源】

Codeforces 269D

【简要题意】

在一个笛卡尔坐标系上有n条平行于x轴的线段，这些线段不相交，且在最上方和最下方都有一条无限长的线段。

对于线段u和线段v，v在u的下方，且u和v的x轴坐标范围有交，且在u和v之间没有可以途径的线段时，那么$f(u,v)$为u和v的x轴坐标范围的交的大小，否则$f(u,v)=0$。

找出一条从上到下的路径，使得途径的最小的f值最大，求出最大的f值。

数据范围：$n \le 10^5$

【解题思路】

因为线段是从上到下一次经过的，所以只需要考虑相邻的两条线段间的转移，对于每条线段计算出最优的f值。

现在需要找转移来源，转移一定来自与有交的线段，所以将线段从上到下依次处理时，只要暴力在线段树中遍历就可以找到可能的转移来源，然后通过判断它们的交在线段树中是否是纯色就可以判断是否合法，转移完成后在线段树中用那条线段覆盖。这样线段树中每次最多只会增加3条线段，所以复杂度是有可靠保证的。

时间复杂度$O(n\log n)$，空间复杂度$O(n\log n)$。

63 Wall Bars

【题目来源】

Codeforces 268D

【简要题意】

Wall Bar的建造要求如下：

（1）建筑物中间那根管子高度为n。

（2）在所有整数高度的地方，恰好有一根水平的横杆从中间的杆子连向四个方向中的某一个预先固定好的杆子上。

（3）如果两根横杆的距离不超过h，且方向相同，那么一个孩子可以从一个一根横杆爬到另一根上。在地上的孩子，可以爬到任何一根高度在$[1, h]$的横杠上。

（4）一个从地面出发的孩子至少能到达一根高度在$[; n-h+1, n]$的横杠。

求有多少种符合要求的Wall Bar的建造方案mod 1000000009。

数据范围：$n \leq 1000$，$h \leq 30$

【解题思路】

由于只有4个方向的杆，而各个方向是本质不同的，且每个高度都有杆，所以可以用前一次出现的位置来表示状态，而最近出现的杆一定是在前一刻出现，所以状态数只有$O(h^3)$，并且可以把杆按前一次出现的时间最小表示来储存进一步缩减状态数。

有一个细节是需要记录最近出现的杆是否已经在之前断掉过，即不能爬到。

时间复杂度$O(nh^3)$，空间复杂;度$O(nh^3)$。

64 Berland Traffic

【题目来源】

Codeforces 267C

【简要题意】

给出一个n个点m条边的无向图网络，每条边有一个流量限制，这个限制是双向的(正着流和反正流的限制一样)，对于除源点1和汇点n 以外的所有点都必须满足流量平衡。

这个流还必须满足一个奇特的性质，对于点u和点v，u到v的所有路径的每条边的流量和必须相等，即对于每条路径L，$\sum_{e \in L} Flow_e$是定值。

求满足这个性质的最大流，以及每条边的流量情况(正负表示方向)。

数据范围：$n \leq 100$，$m \leq 5000$

【解题思路】

为了满足那个性质，可以对每个点赋一个标号ind_i，u到v的边的流量即为$ind_u - ind_v$，这样就可以满足那个性质，并且满足那个性质的流网络一定可以得出这样的标号。

这样我们可以设$ind_1 = 1$，$ind_n = 0$，然后用流量平衡列一个$n-2$元方程组，共$n-2$个方程，这样可以用高斯消元来解出一个初始解。

然后利用这个初始解的ind_i的缩放来得出最优解。

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

65 BerDonalds

【题目来源】

Codeforces 266D

【简要题意】

有一个城市的交通网络由n个节点和m条双向道路组成，每个节点有一个咖啡厅。

现在要选择某个位置（可以是节点，也可以是某条道路上的任意一点）建一个餐馆，使得餐馆到最远的咖啡厅的距离最近，求出这个最短距离。

数据范围：$n \leq 200$

【解题思路】

在点上的情况很容易解决，现在考虑在边上的情况。

设当前边为(u, v)，把所有点按到u的距离排序，枚举从u那一侧走的点集，计算v那一侧点集的最远点，进而算出起点在边上的位置。

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

66 More Queries to Array...

【题目来源】

Codeforces 266E

【简要题意】

维护一个长度为n的数列a，需要支持以下操作m次：

（1）将区间$[l_i, r_i]$赋值成x_i。

（2）计算$\sum_{i=l}^{r} a_i * (i-l+1)^k \ mod \ 1000000007$。

数据范围：$n, m \leq 10^5$，$k \leq 5$

【解题思路】

由于k比较小，所以可以把$(i-l+1)^k$用二项式定理展开，维护类似$\sum a_i * i^k$的信息计算即可。

时间复杂度$O((n+m)k\log n)$，空间复杂度$O(n)$。

67 Roadside Trees

【题目来源】

Codeforces 264E

【简要题意】

在一条直线有n个位置可以种树，自西向东标号1到n。m个月，每个月，每棵树都会长高1米。每个月都会执行一个操作，操作共两种：

（1）在位置p种一棵高度为h的树

（2）砍掉从西向东数的第x棵树，当这个位置的树被砍掉后，倒下的树会占据这个位置，之后这个位置不能再种树。

在做完这个操作后，求出当前树高的最长上升子序列。任何时刻树的高度是不同的。初始没有任何树。

数据范围：$n \leq 10^5$，$m \leq 2 * 10^5$，$h, x \leq 10$

【解题思路】

对于最靠左的10棵树和最矮的10棵树单独维护，剩下的直接求最长上升子序列，修改时暴力重算最靠左或最矮的树的贡献。

时间复杂度$O(m \log n)$，空间复杂度$O(n)$。

68 Rhombus

【题目来源】

Codeforces 263E

【简要题意】

有一个$n * m$的二维非负整数数组A，以及一个约束条件K。找到一对整数(a, b)满足以下条件：

（1）$K \le a \le n - K + 1$

（2）$K \le b \le m - K + 1$

（3）使$\sum_{i=1}^{n} \sum_{j=1}^{m} A_{i,j} * max(0, K - |i - a| - |j - b|)$最大

如果有多个答案输出任意一个。

数据范围：$n, m \le 1000$，$K \le \lfloor \frac{min(n,m)+1}{2} \rfloor$

【解题思路】

考虑直接枚举a, b，然后利用前缀和$O(K)$计算值，这个暴力的复杂度为$O((n - 2K)(m - 2K)K)$。

$$(n - 2K)(m - 2K)K = \frac{(n - 2K)(m - 2K) * 4K}{4}$$
$$(n - 2K) + (m - 2K) + 4K = n + m$$

由基本不等式得

$$(n - 2K)(m - 2K) * 4K \le (\frac{n + m}{3})^3$$

$$\frac{(n-2K)(m-2K)*4K}{4} \leq \frac{(n+m)^3}{108}$$

所以暴力算法的效率就足以通过此题，虽然利用前缀和继续优化可以做到$O(n^2)$，但是这个算法利用了题目性质，更加优美。

时间复杂度$O(\frac{(n+m)^3}{108})$，空间复杂度$O(nm)$。

69 Maxim and Increasing Subsequence

【题目来源】

Codeforces 261D

【简要题意】

给出k个数列，每个数列长度为n，数列中最大的数为B。

对于每个数列，把它复制t遍，得到一个新数列，求新数列的最长上升子序列（严格递增）。

数据范围：$k \leq 10$，$n, B \leq 10^5$，$t \leq 10^9$，$n * B \leq 10^7$

【解题思路】

因为要求严格递增的子序列，所以$t > B$是没有必要的。

考虑DP计算，我们可以维护$f[x]$表示序列最后一个数$\leq x$的最长序列（可以发现$f[x]$是递增的），然后把数列一个一个放入来转移，更新完$f[x]$后继续拿那个值更新$f[x+1]$，$f[x+2]$……直到不能更新为止。

这样在数列上枚举的复杂度为$O(n * min(t, B))$，更新复杂度为$O(nB)$（更新最多nB次）。

时间复杂度$O(knB)$，空间复杂度$O(n + B)$。

70 Maxim and Calculator

【题目来源】

Codeforces 261E

【简要题意】

有一个计算器，这个计算器有两个整数单元，一开始，第一个单元包含数字1，第二个单元包含数字0。

设第一个单元的数字为a，第二个单元的数字为b，那么一次操作可以将第一个单元的数字改成$a*b$，或将第二个单元的数字改成$b+1$。

求有多少正整数$x(l \le x \le r)$，可以在p次操作之内使第一个单元中的数字为x。

数据范围：$2 \le l \le r \le 10^9$，$p \le 100$

【解题思路】

因为操作数只有p，所以对第二个单元格的修改次数不会超过p次，所以x的质因子只会是小于p的质数，那么可以搜索出所有可能的x，设有S个，而$S \le 5000000$，可以接受。

那么就要计算得到这些数的最小操作数，可以用DP解决，设$f[i][j]$表示第二个单元格为i，得到j的最小操作数，数组可以滚存。

时间复杂度$O(pS)$，空间复杂度$O(S)$。

71 Dividing Kingdom

【题目来源】

Codeforces 260E

【简要题意】

一个笛卡尔坐标系上有n个点，用两条平行于x轴的直线和两条平行于y轴的直线把平面分隔成9份，要使这9份点的点数集合为集合a，求一种合法方案。

数据范围：$n \le 10^5$

【解题思路】

先枚举由两条平行于x轴的直线确定的a_i，确定这两条直线，然后再根据已确定的点集来分割两条平行于y轴的直线。

搜索过程中根据可行性加一些剪枝。

时间复杂度$O(C_9^3 * C_6^3 * n)$，空间复杂度$O(n)$。

72 Little Elephant and Broken Sorting

【题目来源】

Codeforces 258D

【简要题意】

给出一个n的排列，以及m个操作，这m个操作依次进行且每个操作都有50%的概率被忽略，求最后的排列的期望逆序对个数。

数据范围：$n, m \leq 1000$

【解题思路】

设最后的数列为A_i，那么就是要求

$$e(\sum_{i=1}^{n}\sum_{j=i+1}^{n}(A_i > A_j))$$

$$= \sum_{i=1}^{n}\sum_{j=i+1}^{n}e(A_i > A_j)$$

于是可以维护$p_{i,j}$为$A_i > A_j$的概率，执行操作时只需要对和修改位置有关的p的值就可以了。

时间复杂度$O(n^2 + nm)$，空间复杂度$O(n^2 + m)$。

73 Liars and Serge

【题目来源】

Codeforces 256D

【简要题意】

有n个人，每个人都说出说真话的人数。

求出有多少种局面可以经过缜密的推理得出m个人显然在说谎。

n是2的整次幂，答案mod 777777777。

数据范围：$n, m \leq 256$

【解题思路】

假如一个人“显然”说了慌，这个人说了A，那么这里说A的人肯定不是A个人。

那么我们可以设计一个DP，$f[i][j][k]$表示现在枚举到数字i，总共用了j个人，有k个人显然在说谎的方案数，这个算法的时间复杂度是$O(n^4)$，不能通过此题，但是由于n是2的整次幂，所以n的种数是很少的，所以可以通过打表来完成。

时间复杂度$O(1)$，空间复杂度$O(n^2)$。

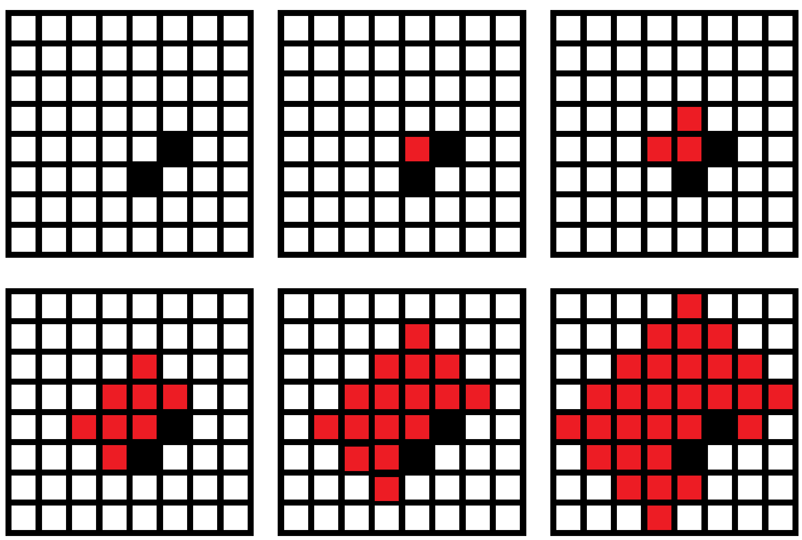
74 Rats

【题目来源】

Codeforces 254D

【简要题意】

在一张$n * m$的方格图中，有一些位置有障碍，有一些位置有老鼠，剩下的是空格。现在需要投掷两个炸弹，每个炸弹会扩散d秒，扩散方式如下图，碰到障碍就会停止扩散。构造一种能炸死所有老鼠的方案。



数据范围：$n, m \leq 1000$，$d \leq 8$

【解题思路】

能炸到的老鼠的数量是$O(d^2)$级别的。

由于每只老鼠都必须被炸死，所以可以先随便选取一只老鼠，在能炸到它的位置放置第一个炸弹，然后再随便找一只没有被炸死的老鼠，在能炸到它的位置放置第二个炸弹，判断是否覆盖了所有老鼠。

时间复杂度$O(nm + d^6)$，空间复杂度$O(nm)$。

75 Printer

【题目来源】

Codeforces 253E

【简要题意】

有n个打印任务，第i个任务在t_i时刻接到，需要打印s_i单位时间，优先级为p_i（p_i互不相同）。

现在我们得知了其中一个任务的完成时间，但不知道这个任务的优先级。我们知道剩下所有任务的优先级，但不知道那些任务的完成时间。

需要求出那个任务的优先级，和所有任务的完成时间，输出任意一种方案，保证至少存在一组解。

数据范围：$n \le 50000$，$t_i, s_i, p_i \le 10^9$

【解题思路】

假如我们得知了那个任务的优先级，我们就可以用一个堆来维护这些任务的优先级模拟，计算出所有任务的完成时间。

而那个任务的优先级越高，完成一定越早，所以可以二分优先级来拟合完成时间。

因为优先级需要互不相同，而优先级的大小并无意义，主要在于大小关系，所以可以选取1，10^9，以及p_i+1这些特殊值来作为参考值，在这些值中二分。

又因为在二分阶段其他任务的完成时间并不重要，所以我们不需要堆，只要记录比二分优先级高的和低的两种状态就行了，在得到一个优先级作为答案后再用堆模拟。

时间复杂度$O(n\log n)$，空间复杂度$O(n)$。

76 Two Sets

【题目来源】

Codeforces 251D

【简要题意】

给出一个大小为n的数集A，把这个集合分成两部分（允许空集），使得两个集合的异或和加起来最大，其次使得较大的异或和尽量大。

数据范围：$n \leq 10^5$，$A_i \leq 10^{18}$

【解题思路】

设原集合的异或和为S，由异或性质可得S中是1的二进制位，分集合后一定是一个1一个0，所以考虑逐位确定S中是0 的位，分集合后是两个1还是0，列异或方程组进行判断，这样可以确定分完集合后的两个异或和的一些位。为了使较大的异或和尽量大，同样可采用逐位确定解决，列异或方程组判断。

同时异或方程组中，有用的元素不会超过64个（二进制位数），所以复杂度有保证。

时间复杂度$O(64n + 64^3)$，空间复杂度$O(n)$。

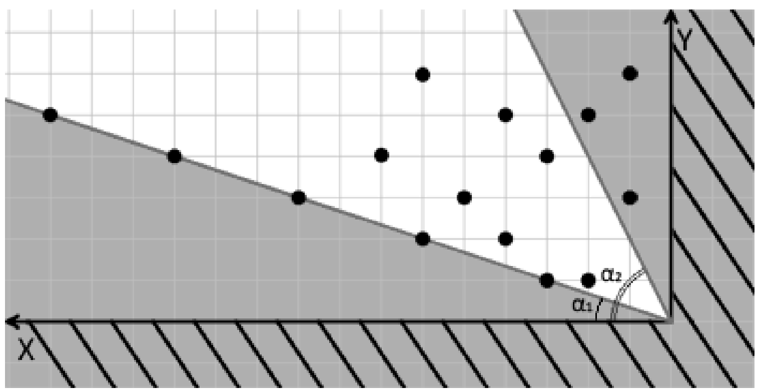
77 Donkey and Stars

【题目来源】

Codeforces 249D

【简要题意】

在一个笛卡尔坐标系上有n个点，从一个点可以看到一个角度范围内的点（如下图），每次可以移动到一个看得见的点，求最多能移动几次。



数据范围：$n \leq 10^5$

【解题思路】

放缩一下坐标系，用树状数组维护模拟拓扑图最长路。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

78 Endless Matrix

【题目来源】

Codeforces 249E

【简要题意】

有一个大小为无穷，构造如下图的矩阵A。

1	2	5	10	17	26
4	3	6	11	18	27
9	8	7	12	19	28
16	15	14	13	20	29
25	24	23	22	21	30
36	35	34	33	32	31

现在有T组询问，每次询问$\sum_{i=x1}^{x2}\sum_{j=y1}^{y2} A_{i,j}$的后十位，如果超出十位则前面再多输出三个点。

数据范围：$T \le 10^4$，$x1, x2, y1, y2 \le 10^9$

【解题思路】

首先要确定是否超出十位，那么就需要在几个不同的模域下，求出来比较。

在模域下计算很简单，可以把询问矩阵分成四个前缀矩阵来计算，然后就是一些等差数列的简单计算了。

时间复杂度$O(T)$，空间复杂度$O(1)$。

79 Piglet's Birthday

【题目来源】

Codeforces 248E

【简要题意】

有n个架子，每个架子上都有一些（或没有）蜜罐。初始时所有蜜罐都是装满蜜的，第i个架子上有a_i个罐子。

Winnie一共去了q次架子，第i次Winnie会先去第u_i个架子，拿走k_i个蜜罐，把这些蜜罐中的蜜吃掉，然后把这些蜜罐都放到第v_i个架子上。当Winnie拿走蜜罐时，他会在第u_i个架子上所有k_i个蜜罐的集合中等概率选择一个集合，然后把集合中的k_i个蜜罐拿走。

Winnie想知道，每次操作后，架子上所有蜜罐都被吃完的架子的期望个数是多少。

数据范围：$n, q \leq 10^5$，$a_i \leq 100$，$k_i \leq 5$

【解题思路】

直接维护$f[i][j]$为第i个架子上还剩j个满罐子的期望，移动罐子后的转移用组合数完成，组合数也直接预处理成实数形式。

时间复杂度$O(nak + mak)$，空间复杂度$O(nak)$。

80 Colorado Potato Beetle

【题目来源】

Codeforces 243C

【简要题意】

在一块大到可以认为是无限的田地上，有许多害虫从无限远处扩散过来，当扩散到有杀虫剂的地方才会停止。

现在从$(0,0)$开始喷杀虫剂，共n次操作，每次操作都将描述为$U/D/L/R\,x$，即朝某个方向一直碰洒x单位距离。求最后会有多少面积的田地可以避免被害虫侵蚀。

数据范围：$n \leq 1000$，$x \leq 10^6$

【解题思路】

把田地离散化之后，直接用$floodfill$广搜解决。

时间复杂度$O(n^2)$，空间复杂度$O(n^2)$。

81 Numbers

【题目来源】

Codeforces 241D

【简要题意】

给出一个n的排列，在这个排列中选择一个子序列，使得满足以下条件：

（1）所有数异或起来是0。

（2）把所有数依次接起来形成的十进制数$mod\, p = 0$。

输出任意解，或判断无解。

数据范围：$n, p \leq 50000$

【解题思路】

由于是一个n的排列，所以可以保证在n比较大的时候一定存在1～31这些数，而实际上只要这31个数就可以构造出解。

可以从概率的方面来尝试证明，题中的两个条件的值都是近随机分布的，所以在这31个数中将有2^{31}种子序列，在题目给定的范围内是可以良好运作产生解的。

具体构造解只需要一个DP就可以完成$f[i][j][k]$表示计算到第i个，异或和为j，十进制数$mod\, p$为k的方案。

时间复杂度$O(32^2p + n)$，空间复杂度$O(32^2p + n)$。

82 Road Repairs

【题目来源】

Codeforces 240E

【简要题意】

给出一张n个点m条边的有向图，对于给定的根，求最小树形图，即求一个最小的子图使得根能到达所有点。

所有边的边权为0或1，需要输出方案。

数据范围：$n, m \leq 10^5$

【解题思路】

最小树形图问题有一种算法——朱刘算法，每次贪心选择每个点的最小入边作为图上边，然后进行缩圈，重构图递归子问题。

但是这个算法的朴素实现的复杂度为$O(nm)$，由于需要输出方案，所以空间消耗也是$O(nm)$，但是在边权为0或1的条件下，对于随机的图可以跑出非常高的效率。

当然朱刘算法存在$O(m \log n)$的实现，可以完美解决这题。

时间复杂度$O(nm)$，空间复杂度$O(nm)$。

83 Torcoder

【题目来源】

Codeforces 240F

【简要题意】

给出一个长度为n的字符串S，进行m次操作，每次操作会选择一个区间$[l_i, r_i]$，如果这个子串可以通过重排表示为回文串，那么把这个区间内的字符进行重排，使其成为字典序最小的回文串。

输出m次操作后的串。

数据范围：$n, m \leq 10^5$

【解题思路】

判断一个字符集是否可以排列成回文串，只需要判断出现次数为奇数的字符是否唯一或不存在即可，所以只需要维护26棵线段树就可以完成对字符集的判断。由于要排成字典序最小的回文串，所以重排后的字符会连成连续的一段，只需要26次区间覆盖操作就可以完成。最后输出完整串时，把标记全部下传即可。

时间复杂度$O(26m\log n)$，空间复杂度$O(26n)$。

84 Meeting Her

【题目来源】

Codeforces 238E

【简要题意】

一张n个点m条边的图表示一个城市，有k条公交路线，每条公交线路会随机选择一条从起点到终点的最短路。

现在一个人想从a坐公交车到b，当这个人在站台时，会随机乘上一辆到这个站台的车，上了车后他只知道这辆车是哪条路线，而不知道这辆车行驶到了哪里。

求在最坏情况下，这个人是否能在有限的时间里到达目的地，如果能到达，求出最坏时间。

数据范围：$n, k \leq 100$

【解题思路】

先求出每条路线的必经点，因为在最坏情况下只有这些地方可以上车。然后就是不停地进行DP转移，假如在有限时间内能到目的地，那么不会到达超过n个点，所以只需要DP做n个周期就可以判断。

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

85 Cyclical Quest

【题目来源】

Codeforces 235C

【简要题意】

对与给定的母串S，做m次询问，每次询问S有多少子串周期性同构于串s_i，周期性同构的定义为经过周期性旋转后相同。

数据范围：$|S| \le 10^6$，$m \le 10^5$，$\sum |s_i| \le 10^6$

【解题思路】

为了解决周期性同构问题，可以把s_i复制倍长后，求$|S|$有多少后缀的$|s_i|$长度的前缀在新的s_i中出现过，也等价与把s_i所有周期性同构的串在$|S|$中求出现次数。

那么用后缀自动机可以较容易地解决这个问题，值需要把新的s_i串在后缀自动机上进行匹配，然后每次把完成匹配的后缀数量计算一下即可，同时还需要注意重复计算的问题，用lazy标记可以完成。

时间复杂度$O(|S| + \sum |s_i|)$，空间复杂度$O(|S|)$。

86 Number Challenge

【题目来源】

Codeforces 235E

【简要题意】

设$d(n)$表示n的约数个数。

求

$$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c} d(i*j*k)\ (mod\ 1073741824)$$

数据范围：

【解题思路】

首先有一个结论

$$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c} d(i*j*k) = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c} \lfloor\frac{a}{i}\rfloor\lfloor\frac{b}{j}\rfloor\lfloor\frac{c}{k}\rfloor(gcd(i,j)=gcd(i,k)=gcd(j,k)=1)$$

这个结论可以从每种约数存于哪些$d(i*j*k)$中来证明，接下来就是计算这个式子。

枚举i以及$d = gcd(j,k)$，然后计算时乘上$\mu(d)$，枚举j,k时的复杂度为$O(n\ln n)$（调和级数）。

中间需要计算gcd来暴力满足$gcd(i,j)=gcd(i,k)=gcd(j,k)=1$，可以预处理$gcd$来优化。

时间复杂度$O(n^2\ln n)$，空间复杂度$O(n^2)$。

87 Figure Eight

【题目来源】

USACO OPEN 13

【简要题意】

给出一张$n * n$的格点图，有一些位置有障碍，要在这个格点图上画一个“8”，上面的框要比下面的框窄且不经过障碍。

求最大的“8”的大小，大小定义是上框面积和下框面积的乘积。

数据范围：$n \leq 300$

【解题思路】

可以递推求出$u[i][j][k]$和$d[i][j][k]$为第i行，左右为j和k的最大上框和下框（以第i行为底和顶），然后对于每行用一个区间DP计算$f[j][k]$为左右范围在$[j,k]$之内的最大上框，然后拼上相应的下框更新答案。

时间复杂度$O(n^3)$，空间复杂度$O(n^3)$。

88 Photo

【题目来源】

USACO DEC 13

【简要题意】

有n头奶牛排成一排，标号为1到n。小明拍了m张照片，照片i包含了标号从a_i到b_i的奶牛，每张照片中恰有一头奶牛有斑点。

问至多有多少头奶牛有斑点，无解输出-1。

数据范围：$n \leq 2 * 10^5$，$m \leq 10^5$

【解题思路】

设A_i为第i头牛是否有斑点，有则为1，没有则为0，设$B_i = \sum_{j=1}^{i} A_j$。

如果$[l, r]$范围内用正好一头牛，那么$B_r - B_{l-1} = 1$，即$B_r - B_{l-1} \leq 1$，$B_r - B_{l-1} \geq 1$，现在要使得$B_n - B_0$最大。

这是经典的差分约束问题，可以用SPFA最短路算法解决，首先需要判断负环来确定是否无解，判负环用栈维护的SPFA效果较优。

时间复杂度$O(m \log n)$，空间复杂度$O(n + m)$。

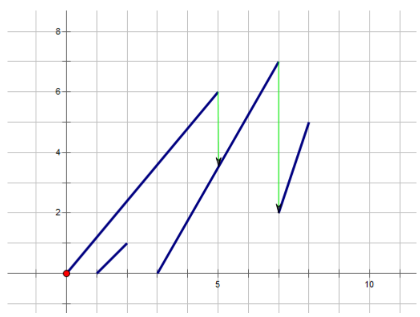
89 Hill Walk

【题目来源】

USACO MAR 13

【简要题意】

在一个平面上有n条线段，描述为$((x_1, y_1), (x_2, y_2))$（保证线段互不相交），从坐标原点$(0, 0)$沿线段往上走，若走到线段端点之后便会掉下来（如下图），求最后会走过几条线段。



数据范围：$n \le 10^5$，$0 \le x, y \le 10^9$

【解题思路】

由于线段的数量不多，所以可以依靠模拟来计算走过的条数，那么现在的问题就是从一个点落下后找到第一条碰到的线段。

由于线段互不相交，所以可以用扫描线的方式维护，用平衡树维护线段的上下关系，询问时在平衡树上二分（遍历）查找即可。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

90 Cows in a Skyscrape

【题目来源】

USACO Mar 12

【简要题意】

电梯的承重为W，n头牛的重量分别为C_i，求最少要多少次电梯才能载完所有牛。

数据范围：$n \le 18$，$C_i \le W \le 10^9$

【解题思路】

这题是PA2014 Pakowanie的弱化版，我得出了以下几种做法：

算法一：搜索

实际测试时搜索的效率最优，配合一些类似状压的剪枝效果极佳。

算法二：状压DP $O(2^n n^2)$

设$f[i][j]$为用到第i个电梯，牛的状态为j，还剩余的最大空间。

算法三：FWT $O(2^n n^2)$

预处理一次电梯可以运送哪些牛的子集，然后对这些子集做位运算“或”的卷积，当卷出全集时就得到了答案。

位运算卷积就是Fast WalshHadamard Transform算法，做一次的复杂度为$O(2^n n)$。

时间复杂度$O(2^n n^2)$，空间复杂度$O(2^n)$。

91 First!

【题目来源】

USACO DEC 12

【简要题意】

给出n个字符串s_i，设总长度为L，求有多少字符串可以在某种字典序表下成为字典序最小的串。

数据范围：$n \le 30000$，$L \le 3 * 10^5$

【解题思路】

对n个串建立字典树，当一个串要成为字典序最小时，它在字典树上路径的每个节点都要是最小的，这样就得到了一个关于字符的字典序约束关系，这个可以用传递闭包判断是否有解，传递闭包用位运算$floyd$实现。

时间复杂度$O(26L + 26^2 n)$，空间复杂度$O(n + 26L)$。

92 Tower of Hay

【题目来源】

USACO DEC 09

【简要题意】

有n堆干草，第i堆干草的宽度为w_i，高度均为1。这些干草必须按顺序一堆一堆叠上去，不能隔层堆叠，低层的宽度必须大于高层，求干草堆的最高高度。

数据范围：$n \le 10^5$，$w_i \le 10^4$

【解题思路】

在所有合法的堆叠方式中，最高的方式的最底层一定可以是最窄的，所以可以利用这个单调性优化DP，维护一个单调队列转移。

时间复杂度$O(n)$，空间复杂度$O(n)$。

93 Cleaning Up

【题目来源】

USACO MAR 09

【简要题意】

有n头奶牛排队进食堂，奶牛需要的食物有m种，每头奶牛需要其中一种。

每次可以选择队首的若干头奶牛进食堂，若它们需要K种不同的食物，则需要消耗K^2的时间。

求最小耗时。

数据范围：$m \le n \le 40000$

【解题思路】

假如$K \ge \sqrt{n}$，那么是肯定不优的，消耗的时间还不如让奶牛一头一头进来，所以$K < \sqrt{n}$。

那么设$f[i]$为让前i头牛进入食堂消耗的最少时间，然后维护$\sqrt{n}$个指针来确定转移来源，转移来源的位置是单调的，所以更新指针有均摊复杂度保证。

时间复杂度$O(n\sqrt{n})$，空间复杂度$O(n)$。

94 Cow Neighborhoods

【题目来源】

USACO OPEN 08

【简要题意】

n只奶牛结成了几个“群”。每只奶牛都有一个不同的坐标。两只奶牛是属于同一个群的，当且仅当至少满足下列两个条件之一：

（1）两只奶牛的曼哈顿距离不超过C，即$|x_1 - x_2| + |y_1 - y_2| \le C$。

（2）两只奶牛有共同的邻居。即存在一只奶牛k，使k分别与这两只奶牛同属一个群。

计算有多少个牛群，以及最大的牛群里有多少只奶牛。

数据范围：$n \le 10^5$，$|x|, |y|, C \le 10^9$

【解题思路】

旋转坐标系，把曼哈顿距离转成两维坐标距离的Max，这样可以把牛先按x坐标排序，用set维护牛的y坐标来进行并查集合并。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

95 Triangle Counting

【题目来源】

USACO OPEN 08

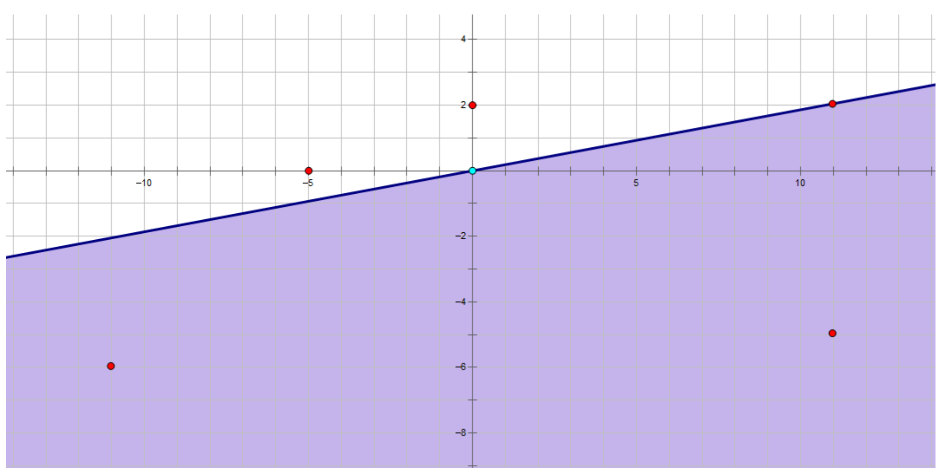
【简要题意】

给出一个n个点的整数点集，求以这个点集中的点为顶点的三角形，有多少包含了原点。

数据范围：$n \leq 10^5$

【解题思路】

问题可以转化为求有多少三角形不包括原点，假如一个三角形不过原点，那么如果做一条过原点和三角形上极角最小的点的直线，另两个点一定在直线的另一侧，如下图。



那么我们可以枚举三角形中极角最小的点，利用单调性维护那条直线，若直线的一侧有K个点，那么就可以构成C_K^2个不包含原点的三角形，最终答案就是$C_n^3 - \sum C_K^2$。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

96 Land Acquisition

【题目来源】

USACO MAR 08

【简要题意】

有n块土地，每块土地的长为W_i，宽为L_i，如果一次性购买一批土地，则需要的花费为$Max(W_i) * Max(L_i)$。

求买完所有土地的最小花费。

数据范围：$n \le 50000$，$W_i, L_i \le 10^6$

【解题思路】

如果有两块土地a和b，$W_a \le W_b$，$L_a \le L_b$，那么a这块土地显然是对答案没有任何贡献的。

所以可以认为，把所有土地按W_i排成升序后L_i即为降序，那么容易得到DP转移$f_i = Min(f_{j-1} + W_i * L_j)$，然后可以用斜率优化来做到$O(n)$转移。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

97 Toys

【题目来源】

USACO NOV 08

【简要题意】

有n天，第i天需要的玩具数T_i，有一个每天以Tc美元卖出商品的玩具店。

除了购买外，还有两种消毒方式：第1种方式需要收费C_1美元, 需要N_1个晚上的时间；第2种方式需要收费C_2美元，需要N_2个晚上的时间。

求最小花费。

数据范围：$N_1, N_2 \le n \le 10^5$，$T_i \le 50$，$C_1, C_2 \le 60$

【解题思路】

三分强制购买的玩具数量（打表可以发现具有三分性质），然后用最优的方式去安排消毒方式。

当已经确定了强制购买数量时，就可以贪心了，尽可能多选收费较小的消毒方式。

时间复杂度$O(n \log nT)$，空间复杂度$O(n)$。

98 Fence

【题目来源】

USACO DEC 08

【简要题意】

给出n个点，求一个最大的子集，使得这些点构成一个凸包。

数据范围：$n \leq 300$

【解题思路】

可以采取DP，设$f[i][j]$为前两个点分别为i和j的最大凸壳来求，转移则为枚举下一个点k，转移到$f[j][k]$，但是这样DP必须枚举起点才可以保证形成完整凸包，所以这样暴力的时间复杂度是$O(n^4)$的。

然后发现转移的时候我们需要使得两个向量的偏转角$< 180°$，所以转移方向其实是随向量角度单调的，所以可以预处理每个点左右两边的向量来进行单调的转移，来优化掉k那维的枚举。

时间复杂度$O(n^3)$，空间复杂度$O(n^2)$。

99 Best Cow Line

【题目来源】

USACO DEC 07

【简要题意】

给出一个长度为n的字符序列，每次可以从序列头或者序列尾选择一个字符放到新序列的末尾。求字典序最小的新序列。

数据范围：$n \le 30000$

【解题思路】

贪心。设正序的串为S_1，倒序的串为S_2，那么每次从字典序较小的串中取字符。

那么就只需要比较两个串的字典序，Hash求出LCP（最长公共前缀），后比较后一个字符即可。

时间复杂度$O(n \log n)$，空间复杂度$O(n)$。

100 Cow Patterns

【题目来源】

USACO DEC 05

【简要题意】

给出一个长度为n的串，字符集大小为S。对于一个长度为K的模版串，求在原串中有多少子串满足，字符的相同关系和大小关系和模版串一样，并输出这些子串的位置。

数据范围：$n \le 10^5$，$K \le 25000$，$S \le 25$

【解题思路】

问题可以转化为，求有多少子串最小表示后（仍保留大小关系），和模版串相同。

对于这类问题常用KMP算法解决，在KMP比较时，对于新加入的字符，只需比较它之前有多少字符和它一样，有多少字符比它小这两项就可以了。考虑前面的串已经比较完成，那么这两个数值就可以确定这个字符最小表示后的情况。

计算字符量可以预处理前缀和$O(n) - O(1)$完成。

时间复杂度$O(n + K)$，空间复杂度$O(n + K)$。