|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 试题编号 | 名称 | 题意 | 算法 | 时空复杂度 |
| 8800 | Tournament-graph | 要构造一个有N个结点的竞赛图，使得对任意两个结点u和v（u≠v），从u到v的最短距离不超过2。  竞赛图就是基图为无向完全图的有向图。 | 构造(看题解) | 时间O(n^2)  空间O(n^2) |
| 8768 | Three Swaps | 饲养员Xenia有n只站成一排的马。每只马都有它自己的独一无二的号码。最开始的时候，从左数第i只马编号为i。她会给马命令。每个命令是一对数l，r。命令l，r意思是 [l，r]位置内的马反向排列　Xenia最多最多给出三个命令。现在你已经知道了最后马的编号序列，请给出训练时Xenia的命令是什么。 | 搜索 | 时间O(1)  空间O(n) |
| 8801 | Land Acquisition | Farmer John要买N块长方形的土地，每块土地长为Li，宽为Wi。  如果一次性购买几块土地，总价钱就是这几块土地的max{Li}\*max{Wi}。Farmer John想通过分次购买，花最少的钱把所有土地都买下来，问他最少要花多少钱。 | 动态规划 | 时间O(nlgn)  空间O(n) |
| 8798 | Jeff and Removing Periods | jeff能有一个长度为n的序列 a[1], a[2], ..., a[n].询问区间内是否有一种颜色成等差序列和有多少种颜色。 | 线段树 | 时间O(nlgn)  空间O(n) |
| 8763 | Two permutations | 你有两个各包含n个元素的排列p和q，和m个由l1，r1，l2，r2组成的询问。每次询问在p中位置在[l1,r1],在q中位置在[l2,r2]中的数的数量。 | 主席树 | 时间O(nlgn)  空间O(nlgn) |
| 8762 | First! | 给定一些串询问哪些串在可以存在至少一种改变字母字典顺序后，成为字典序最小的串。 | Trie树 | 时间O(n)  空间O(n) |
| 8784 | White, Black and White Again | 已知，接下来会发生w件两两不同的好事和b件两两不同坏事，每天至少发生一件事，每天要么全部发生好事要么全部发生坏事。这n天会先有若干天发生好事，再有若干天发生坏事，再有若干天发生好事。要求统计事件发生的方案数 | 暴力 | 时间O(n^2)  空间O(n^2) |
| 8746 | BerDonalds | 给定一个无向带权联通图，求图的直径。 | Floyd+二分 | 时间O(n^3logn)  空间O(n^2) |
| 8805 | Sereja and Squares | 给定一个括号序列的一些左括号询问有多少种括号序列 | 动态规划 | 时间O(n^2)  空间O(n) |
| 8810 | Yaroslav and Algorithm | 描述一个算法。  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，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。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。  有一个n个正整数的集合，需要一个这样的算法，能够使每一个数加1。 | 构造+智商 | 时间O(1)  空间O(1) |
| 9195 | Little Elephant and Broken Sorting | 有一个p1,p2,p3…pn的排列。进行m次操作，第i次操作会交换第ai个数和第bi个数，每次操作是会有一半几率不操作。求出在操作完后的排列中的逆序对的数量的期望个数。 | 数学期望 | 时间O(n^2)  空间O(n^2) |
| 9189 | Distinct Paths | 有一个n\*m的木板，一些块已经被涂上给出的k种颜色中的一种。需要把每个没涂色的块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块。路径只能向右或向下走。 | 搜索+位运算 | 时间O(?)  空间O(nm) |
| 9231 | PE lesson | 有一排学生互相丢几次球。每次丢球时，老师会选择任意两个不同的学生，被选中的学生互相向对方丢球。所有学生都只能进行最多两次丢球。询问在这之后，一共有多少种不同的持球方式。 | 数学 | 时间O(n)  空间O(n) |
| 8738 | Rhombus | 你现在得到了一个大小为n×m的表。在第i行与第j列的交叉处，有一个非负整数ai,j。此外，你还得到了一个非负整数k。找到一对满足下列条件的整数对(a,b)  1.k<=a<=n-k+12.k<=b<=m-k+1  3.让我们在k<=x<=n-k+1 且 k<=y<=m-k+1 的范围内定义函数http://www.tsinsen.com/RequireFile.do?fid=6JgafLY4的最大值是mval。对于我们要找的整数对必须满足下列等式f(a,b)=mval。 | 暴力，信就有 | 时间O(n^3)  空间O(n^2) |
| 8739 | Yaroslav and Arrangements | 数列a1，a2，… ， ar为良好的，当它满足：  1.| a1 – a2 | = 1，| a2 – a3 | = 1，…，| ar-1 – ar | = 1，| ar – a1 |= 1  2. http://www.tsinsen.com/RequireFile.do?fid=gnbbgMHq  数列b1，b2，… ， br为优秀的，当它满足：  1. 数列中的元素不下降  2. 满足 1 ≤ r ≤ n 、 1 ≤ bi ≤ m  3. 通过重排数列中的元素可以得到至少一个至多k个不同的良好数列。询问知道有多少个不同的优秀数列。 | 动态规划 | 时间O(n^4)  空间O(n^2) |
| 8740 | Reclamation | 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。 | 并查集 | 时间O(nlgn)  空间O(n) |
| 8802 | Playing with String | 两个人正在玩一个游戏。两个人轮流行动，不能操作的人输。游戏开始前裁判会在方格纸上写下一个字符串，每个格子包含一个字母。一个人的操作分这么几步：1.这个人选择一张纸，我们称上面写着的字符串为t。2.这个人选择一个i(1<=i<=|t|)使得存在一个正整数k(0<i-k,i+k<=|t|)满足t[i-1]=t[i+1] and t[i-2]=t[i+2] ... t[i-k]=t[i+k]3.这个人以把这张纸的第i个字母的两侧撕开使得这张纸分成3份:t[1~i-1],t[i~i],t[i+1,|t|]。  询问胜者是谁。假如先行动的人胜，那么你还需要输出可以帮助这个人获胜的第一步行动选择的位置。如果有多个输出最小的那个。 | Sg函数 | 时间O(n)  空间O(n) |
| 8750 | More Queries to Array... | 你有一个包含n个整数a[1],a[2],...,a[n]数组。有以下两种操作。  1.将闭区间[l,r]中的元素赋值为x。  2.计算并输出和http://www.tsinsen.com/RequireFile.do?fid=fm56Beeb。 | 数据结构 | 时间O(nlgnk^2)  空间O(nk) |
| 8769 | Tennis Rackets | 这将是一种正三角形的球拍。你要在这个框架上穿线。正三角形的每条边上有n个小孔，把该边分成了等距的n+1段。线不能穿过每条边上离每个顶点最近的m个小孔。球拍网需要连成一个钝角三角形，三个角分居正三角形框架三条不同的边上。询问有多少种可行的三角形球拍网设计方案。 | 暴力 | 时间O(n^2)  空间O(1) |
| 8803 | Have You Ever Heard About the Word? | 一个重复块由一个字符串与自身连接而成。你有一个由拉丁字符组成的字符串。每一步你要找到它的子串中最短的重复块，如果有多于一个，你必须选择最左边的那个。然后删除其中的一部分。重复以上步骤直到字符串中不存在重复块。输出做完所有操作后的字符串。 | Hash+暴力 | 时间O(nlgn)  空间O(n) |
| 8809 | Number Challenge | 定义d(n)为n的约数个数。现在有三个数a,b,c。对下面式子求和http://www.tsinsen.com/RequireFile.do?fid=D7fdgLL8 | 数学 | 时间O(n^2)  空间O(n^2) |
| 8807 | Tape Programming | 有一种编程语言，在这个语言下一个程序是由数字和”<”,”>”构成的非空串。程序运行时有一个指针。最开始指针的指向最左字符，移动方向为向右。我们重复以下操作直到指针指向串外：1、如果指针指的位置是一个数字，输出这个数字，然后将指针沿着原来移动方向移动，同时将原来的数字减一。如果原来的数字为0则删除这个数字，串的长度减一；2、如果指针指的位置是”<”或”>”，那么指针的移动方向对应得改为向左或向右（与符号的尖角方向相同），接着指针沿着新的移动方向移动。如果新的位置也是”<”或”>”，则删除原来的”<”或”>”字符。任何时刻如果指针指向了串外，程序就结束运行。  现在我们有一个由n个由"<", ">"和数字构成的串s1, s2, ..., sn ，你需要回答q个询问。每个询问会给你两个数l,r，如果把sl, sl + 1, ..., sr 看出一个单独的程序，问你每个数字会被输出多少次。 | 链表 | 时间O(n)  空间O(n) |
| 8787 | Ciel and Flipboard | 有一个n行n列的板子，每个格子上有一个数字。n是一个奇数，设x=(n+1)/2。你可以选择一个x行x列的子矩阵，并将其中的所有元素乘-1。你可以使用这个操作任意多次。最大化板子上的数字和。 | 搜索 | 时间O(?)  空间O(n^2) |
| 8808 | Rotatable Number | 有”可旋转数”形如142857  　142857 \* 1 = 142857;  　142857 \* 2 = 285714;  　142857 \* 3 = 428571;  　142857 \* 4 = 571428;  　142857 \* 5 = 714285;  　142857 \* 6 = 857142  将“可旋转数”推广到了任意进制b。求最大的b(1 < b < x)，满足在b进制下存在一个长度为n的正“可旋转数”（允许有前导零）。 | 结论题 | 时间O(nlgn)  空间O(1) |
| 8796 | Dima and Game | 纸上有n对整数(l[i],r[i])(1<=l[i]<r[i]<=p)。玩家轮流进行操作。轮到自己时，进行下面的操作：  1.选择第i对数(1<=i<=n)，满足r[i]-l[i]>2;  2.将第i对数替换为(l[i]+floor((r[i]-l[i])/3),l[i]+2\*floor((r[i]-l[i])/3) )或者(l[i] , r[i]-floor((r[i]-l[i])/3) )。floor(x)表示向下取整。  不能进行操作的玩家则输。  输出有多少种数使得的先手必胜 | Sg函数 | 时间O(nlgn)  空间O(n) |
| 8791 | Maxim and Calculator | 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。 这个计算器支持一以下两种操作：  1.假设第一个单元的数字为a，第二个单元的数字为b，那么将第二个单元的数字改成b+1。2.假设第一个单元的数字为a，第二个单元的数字为b，那么将第一个单元的数字改成a\*b。询问有多少个正整数x(l<=x<=r)满足， 存在一种方式从计算器初始状态开始，操作不超过p步之后使得第一个单元中的数字为x。 | 搜索 | 时间O(?)  空间O(?) |
| 8743 | Dividing Kingdom | 一个名叫Flatland的国家可以看成一个无穷的二维平面。Flatland有n个城市，每个城市可以看做平面上的一个点。国王Circle IV统治着Flatland。Circle IV有9个孩子，他想让每个孩子分得Flatland的一部分。他想画4条不重合的直线，其中2条与x轴平行，另2条与y轴平行。于是，Flatland被分成了9部分，每个孩子将得到这其中的一块。找到这样的4条直线，使得Flatland被这些直线分成9部分后，第i个孩子可以得到城市个数为ai的一部分。 | 搜索 | 时间O(?)  空间O(?) |
| 8756 | Wall Bars | 有一项在儿童公园内建设Wall Bar的订单。  1、我们定义单位长度，建筑物中间那根管子高度为n。  2、在高度为1、2...n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之间的横杠上。在Manao的建筑物上，一个从地面出发的孩子至少能到达一根高度在n-h+1、n-h+2...n的横杠。  　　Manao想知道有多少种设计方案满足上述要求。 | 动态规划 | 时间O(nh^3)  空间O(nh^3) |
| 8795 | Xenia and Dominoes | 一个拼图是一个3 × n的桌子，除掉一些禁止块，并包含多米诺骨牌。一个拼图被称作合法当它符合以下条件：  每个多米诺骨牌覆盖正好两个非禁止块；  没有两个多米诺骨牌覆盖桌子上的同一块区域；  有且仅有一个非禁止块没被任何多米诺骨牌覆盖。  要完成一个拼图，你需要用若干步把空格子从开始位置移动某个指定位置。一步移动是在保证拼图合法的情况下，把一个多米诺骨牌移到空格子里。横向的多米诺骨牌只能横向移动，纵向的多米诺骨牌只能纵向移动。你不能旋转多米诺骨牌。  如果她把多米诺骨牌放在桌子上，能有多少种不同的合法的拼图。同时，要求圆圈标记的格子没有被覆盖。这个拼图还必须至少能移动一次。统计上述拼图的种数。 | 动态规划 | 时间O(n)  空间O(n) |
| 8767 | Cow Tennis Tournament | Farmer John正在举办一场网球锦标赛，有n只奶牛参与。每只奶牛有一个能力值si，任两只奶牛的能力值互不相同。任意两只奶牛在锦标赛中只交手一次，等级高的总是击败等级低的。  他准备改变一些比赛结果。他会改变k次，每次选择两个整数ai,bi(ai<bi)并改变能力值在ai到bi范围内的奶牛相互之间的比赛结果。  询问出现多少个胜负三元环。 | 线段树 | 时间O(nlgn)  空间O(n) |
| 9199 | Lucky Tickets | ticket是一个数列。  对于一个ticket，如果我们在它一些数的左边或者右边加入一些运算符号(“+”, “-”, “\*”)和括号使得最终结果等于k,那么这个ticket就是k-lucky ticket。  求m个k-lucky ticket | 搜索 | 时间O(?)  空间O(?) |
| 9279 | Maxim and Increasing Subsequence | 数列a给定如下：  1、数列的长度为n\*t；  2、ai=b((i-1) mod n)+1 (1≤i≤n\*t)，求出数列的最长上升子序列。 | 动态规划 | 时间O(n\*maxb)  空间O(n) |
| 9235 | Polygon | 输出每个角角度都相同，每条边长度都不相同的凸n边形 | 乱搞+智商 | 时间O(n)  空间O(1) |
| 9233 | Numbers | 有1,2..n的排列a1,a2,……,an。删除一些整数,使得结果序列满足以下三个条件:  1.由此产生的序列不是空的;  2.序列中所有数异或和等于0;  3.如果把所有数按十进制从前往后依次无间隔地写在一行形成一个大的十进制数，这个数将会被p整除。  找到一种方法来满足上述条件。 | 动态规划 | 时间O(nlg^2n)  空间O(nlg^2n) |
| 9221 | Candies Game | 一开始有n个编号从1到n的箱子，每个箱子有一定数量的糖果在里面。用一个数组a来描述，ai表示箱子i里面的糖果数目。  这个游戏的目的是将所有的糖果放入恰好两个箱子里。剩下的n-2个箱子里面必须一个糖果都没有。你可以执行一系列操作或不执行任何操作。每一次可以选择两个不用的箱子i，j。不妨假设ai<=aj。然后你从箱子j拿ai颗糖果放入箱子i中。  给出一系列操作完成游戏。 | 乱搞 | 时间O(nlgn)  空间O(n) |
| 9173 | Positions in Permutations | P是n个互不相同且不超过n的正整数的一个排列。我们设排列P的第i个元素为P[i]，n为排列的长度。  我们称排列中的第i个位置是完美的，当且仅当|P[i]-i|=1。  请求出长度为n的而且完美的位置数刚好为k的排列数是多少。 | 动态规划 | 时间O(n^2)  空间O(n^2) |
| 9167 | Best Cow Line | 给定一个字符串，每次可以从头或尾取一个字符，求这样生成的字典序最小的字符串 | 后缀数组 | 时间O(nlgn)  空间O(n) |
| 9187 | Cow Patterns | FJ的N只牛中出现了K只爱惹麻烦的坏蛋。FJ给每一只牛一个1..S的编号。现在，FJ已经不记得坏蛋们的具体号码，但是凭他的记忆，他给出一个编号在1..S的“模式串”。原坏蛋的号码如果相同，模式串中他们的号码依然相同。模式串中坏蛋们之间的号码的大小关系也与原号码相同。请找出K个坏蛋的团伙的所有可能性。 | Kmp | 时间O(nm)  空间O(nm) |
| 8765 | Rats | 我们用一个n×m的方格网表示商店的地下室。有些格子被墙占据，其余的格子是空的，其中有一些格字有老鼠。手榴弹只能在没有被墙占据的格子中炸毁。手榴弹爆炸形成的波按照下面的方式扩散。我们假设手榴弹在0时刻爆炸。在这个初始时刻，只有手榴弹炸毁的那个格子被“清除”了。如果在t时刻，某个格子被“清除”了，那么在t+1时刻，那些与这个格子相邻且没有被墙占据的格子也被“清除”了（有些格子可能之前就被“清除”了）。爆炸波扩散恰好d秒，然后它立刻消失。是否可以选择两个格子，使手榴弹在这两个格子中被炸毁，使得所有有睡觉的老鼠的格子都被“清除”。找到这两个格子。 | 搜索 | 时间O(nm)  空间O(nm) |
| 8799 | Friends | 你想给N个朋友拍M张照片。每张照片中有两个人，并且不可以给同一对朋友拍两张照片。每个朋友有一个“好看度”，第i个朋友的好看度为Ai，并且第i和第j个朋友在一起照相时照片的好看度为Ai xor Aj。你想拍M张照片使得这些照片好看度的总和最大。求出最大的总和 | Trie树 | 时间O(nlgn)  空间O(n) |
| 8744 | Piglet's Birthday | Piglet要过生日了，Winnie想送给他一些蜜罐。Winnie的家里有n个架子，每个架子上都有一些蜜罐。Winnie一共去了q次架子；第i次Winnie会先去第ui个架子，拿走ki个蜜罐，把这些蜜罐中的蜜吃掉，然后把这些蜜罐都放到第vi个架子上。当Winnie拿走蜜罐时，他会在第ui个架子上所有ki个蜜罐的集合中等概率选择一个集合，然后把集合中的ki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。 | 动态规划 | 时间O(nmaxa)  空间O(nmaxa) |
| 9174 | Greg and Caves | Greg有一个平板电脑，平板电脑的屏幕是一个由N\*M个像素点组成的矩形，每个像素点可以是黑色或白色。我们设画板的行从上到下编号从1到N。类似地，画板的列从左到右编号为1到M。  Greg认为，如果符合以下条件，那么平板电脑的屏幕就会呈现一个洞。  一）存在一个区间[L,R](1<=L<=R<=N)，使得上至行L、下至行R，每一行都恰有2个黑点，而其他列只有白点。  二）存在一个行号t∈[L,R]，使得：  定义：【中间白点是指同一行处在2个黑点之间的白点，black\_left[i]表示第i行靠左边的黑点所在列，black\_right[i]表示靠右边的黑点所在列】  1）对于任意L<=i<=j<=t，  第i行的任意列的中间白点，第j行的同一列都是中间白点。且black\_left[i]>=black\_left[j]、black\_right[i]<=black\_right[j]  2）对于任意t<=i<=j<=R，第j行的任意列的中间白点，第i行的同一列都是中间白点。且black\_left[i]<=black\_left[j]、black\_right[i]>=black\_right[j]  求能显示一个洞的方案数。  Greg想知道，有多少种方法使得平板上显示一个洞。 | 动态规划 | 时间O(nm)  空间O(nm) |
| 9186 | Random Ranking | 现在请你假想一次考试，这场考试有N个考生。你可以预测这N个考生最终得分的范围，从而预测他的排名。选手们的分数将按照成绩升序排列，也就是说，分数越高，排名越后。你知道第i个考生的得分范围是[Li,Ri]，现在你要预测每个考生得到任何排名的概率。 | 动态规划 | 时间O(n^3lgn)  空间O(n^3) |
| 8755 | The Great Julya Calendar | 有一个写着魔法数字的神圣板块：  取走魔法数字，从中减去一个数字（这个数字必须在魔法数字的数位中出现过）并得到一个新的魔法数字。重复这个操作直到一个魔法数字等于0。　不同的减法序列显然能得到不同的操作数。计算最少需要的操作数来把魔法数字减至0。 | 乱搞 | 时间O(?)  空间O(?) |
| 9230 | Cleaning Up | Farmer John有n头奶牛，m种食物，每头奶牛固定只吃一种食物。现在，n头牛排好队来食堂就餐，FJ每次可以放进来队伍前端任意数量的奶牛，然后同时给它们做食物吃。但是当食堂里面的奶牛吃K种不同的食物时，他需要K\*K个单位的时间来打扫食堂。只有当食堂内所有的奶牛都吃完，并且FJ打扫完食堂之后才能再次放进来其他的奶牛。问当所有的奶牛都就餐完毕之后，FJ最少总共需要花多少时间来打扫食堂。 | 动态规划 | 时间O(nsqrtn)  空间O(n) |
| 9280 | Evil | 在一个二维平面上有n个城市。两个城市之间的距离等于它们之间的曼哈顿距离。哈密尔顿回路定义为所有n个城市的一个排列。哈密尔顿回路的长度定义为排列中相邻城市的距离加上第一个和最后一个城市的距离。请求出给定城市的最长可能的哈密尔顿回路的长度。 | 贪心 | 时间O(n)  空间O(n) |
| 8779 | Tower of Hay | Bessie必须用一垛垛干草建造一个塔从而能够换灯泡。N摞干草在传送带上有序的进入牛栏。第i摞干草有一个为整数的宽度w\_i；所有干草摞有一个单位的高度。Bessie必须用全部的N摞干草去建造那个塔而且必须按照它们到达的顺序放置它们。她可以按照她的愿望放置随意多摞干草在一行上。接下来她可以放置接下来到达的干草摞在上次排成的一行上面来建造新的一行（但不能比下面的行要宽）。直到所有的干草被用完。她必须按干草到达顺序来堆叠它们。 | 动态规划 | 时间O(n)  空间O(n) |
| 8793 | Colorful Stones | 现有两串彩色的石头。每块石头的颜色是红、绿、蓝之一。给你两个字符串s和t。s的第i个（从1开始标号）字符代表第一串的第i块石头的颜色。类似地，t的第i个（从1开始标号）字符代表第二串的第i块石头的颜色。每个字符是'R','G','B'时，对应的石头分别为红色、绿色和蓝色。初始时Squirrel Liss站在第一串的第一个石头上，Cat Vasya站在第二串的第一个石头上。你可以发出下面的指令。每个指令是以下三种中的一种：“红色”、“绿色”、“蓝色”。在发出指令c后，站在颜色为c的石头上的动物将会移动到后面的一块石头上。你不能发出能使某动物走到石头串之外的指令。 (Liss的位置,Vasya的位置)被称为状态。如果从初始状态(1,1)能够通过发出0次或多次指令达到一个状态，我们就称这个状态为“可达”的。请计算出不同的可达状态的数目。 | 动态规划 | 时间O(n)  空间O(n) |
| 8806 | Balance | 给定一个由n个注水容器组成的系统。某几对容器由输水管道相连接。通过管道，你可以将整数升的水在其连接的容器间进行传输（管道都是双向的）。两容器间可能连着多于一根管道。管道的数量为e。每个容器的容积为v升。给定每个容器初始状态的水量a[i]和目标状态的水量b[i]，请求出一个输水方案实现这一目标。输水的总步骤数不能超过2\*n^2。 | 并查集+构造 | 时间O(nm)  空间O(n^2) |
| 8770 | Donkey and Stars | 在夜晚，Donkey会陪着Shrek一起仰望星空。他们坐在木桩上，一边品着茶，一边观赏漫天的繁星。远远望去，夜空挂在屋顶上，位于烟囱的右侧。Shrek的星星在烟囱右侧，Donkey的在左侧。  Donkey每天都会数着星星，所以他知道一共有n颗星星。这一次他决定挑战一下自己。他脑补了一个坐标系：屋顶和烟囱的交点为原点O，OX坐标轴沿着屋顶向左，OY坐标轴沿着烟囱向上。Donkey想象从原点发出两条射线，和OX的夹角分别是α1、α2。现在，他可以从这两条射线的内侧（不包含边界）任意选择一颗星星。之后，他又想象出新的两条射线，从选中的星星出发，和OX方向的夹角也分别是α1、α2。他就这样不断重复着这个步骤，直到无法继续下去。到最后，Donkey得到了一条星星连成的链。你的任务是帮Donkey计算，这条链最多可以有多少颗星星。 | 动态规划 | 时间O(nlgn)  空间O(n) |
| 9208 | Two Sets | 给定n个数使得他们分成两部分以后的异或值之和最大。 | 贪心 | 时间O(nlgn)  空间O(nlgn) |
| 8804 | Google Code Jam | 在每场比赛中，参赛者有几道题目可以做，每道题目有两个部分:简单(small input)和困难(large input)。在你做出简单部分之前是不允许做困难部分的。解出每道题目的简单或者困难部分后，会得到一些分数。参赛者能够立刻知道自己的简单部分的解的提交的结果，但是困难部分的解的正确与否只能在比赛结束公布。最后排名以得分从高到底排序，得分相同按罚时从低到高排序。  Vasya打算去做GCJ，他可以在比赛开始后快速读题。然后对于n道题目，他知道以下5个值。  1、scoreSmall\_i 表示的是做出简单部分之后得到的分数。scoreLarge\_i 指的是做出困难部分后得到的另外分数。2、 timeSmall\_i 表示的是做第i题的简单部分所花的时间。timeLarge\_i表示从简单解法，优化到困难的解法的所需额外时间。3、Vasya一旦写完简单部分的解，就可以1A。然而对于困难部分，他有probFail\_i 的概率FST。并且值得注意的是，如果FST，那么就不会得到困难部分的分数，并且这次提交也不算一个正确解的提交。比赛总共有t时间。读题和提交题的时间忽略不计。并且值得注意的是：Vasya不允许在一道题目的其中一个部分中提交两次解。Vasya想要正确安排时间和解题顺序，然后使得得到的期望分数尽可能高，同时保证期望罚时尽可能少。 | 数学期望+动态规划 | 时间O(nt)  空间O(nt) |
| 8797 | Counting Skyscrapers | 大街上建好了一排摩天大楼。摩天大楼的数量是在 2 到 314!中均匀随机选择的。每座摩天大楼的高度是被独立地随机选择的：对于每个正整数 i，楼层数为 i 的概率为 2-i。如果一座摩天大楼有 i 层，那么它的楼层被编号为 0 到 i - 1。  为了加快中转运输的效率，摩天大楼间修建了一些滑索。一座摩天大楼的第 i 层和另一座摩天大楼的第 i 层之间有滑索当且仅当两楼之间没有摩天大楼有第 i 层。  Alice和Bob想数一数有多少座摩天大楼。  Alice是个严谨认真的人，她想知道摩天大楼数量的准确值。于是她把计数器初始化为 1，从最左边的摩天大楼开始往右走，每次走到一个摩天大楼就把计数器加上 1，直到她到达最右边的摩天大楼。  Bob很没耐心，他想尽快完成任务。于是他把计数器初始化为 1，从最左边的摩天大楼开始利用滑索从一座摩天大楼滑到另一座。因为恐高，每次他会忽略那些高度大于 h 的滑索，然后在剩下的往右滑的滑索中选一个最高的。由于Bob使用滑索时滑得太快以至于他无法数清经过了多少座摩天大楼，所以他直接将计数器加上 2i，其中 i 是他当前所在的楼层编号。他会一直持续这个过程直到他到达最右边的摩天大楼。当Alice和Bob到达最右边的摩天大楼时，他们会比较计数器的值。现在给出Alice的计数器的值或者Bob的计数器的值，请你求出另一人的计数器的值的期望。 | 数学期望 | 时间O(nh)  空间O(1) |
| 9227 | Flights | LiLand 是一个国家，包含 n 个城市。城市的编号从 1 到 n。这里有许多单向的航班让你可以在两个城市之间旅行，但是这些航班的安排使得你一旦离开一个城市就永远不会再回来了。  在以前，每趟航班恰好飞行一个小时，但她想要把某些航班的飞行时间改为 2 小时，使得所有从 1 号城市到 n 号城市的旅行路线都花费相同的时间。 | 最短路 | 时间O(nm)  空间O(m) |
| 8783 | Close Vertices | 你得到了一棵包含n个点的树，树上的每条边有一个非负边权，树上两点间路径的长度是该路径包含的边数，树上两点间路径的权重是指该路径包含的边的边权之和。我们说两点是“相邻”的，当且仅当，存在一条连接该两点的路径，满足该路径的长度小于等于L,且权重小于等于W。统计有多少个点对(u,v)，满足u < v，且u,v是相邻的。 | 树分治 | 时间O(nlgn)  空间O(n) |
| 9234 | Ping-Pong | 在这个问题中，每个时刻您都有一个区间的集合。您每次可以从集合中的区间(a, b)移动到另一个满足c < a < d或者c < b < d的区间(c, d)。您需要判断是否有一种从区间x到区间y的移动方案。 | 并查集 | 时间O(nlgn)  空间O(nlgn) |
| 8745 | Colorado Potato Beetle | Old MacDonald有一个农场和大小为(1010 + 1) × (1010 + 1)平方米的巨大的土豆种植地。这块田地被分成正方形的小块，每块占地一平方米。  Old MacDonald知道科罗拉多土豆甲虫将要入侵，并且会破坏收成。Old MacDonald想在一些土地上喷上杀虫剂。Old MacDonald去了田里，站在田地中间的那个小块中间并将这个格子撒上杀虫剂。现在，他将要通过一系列的移动来在更多田地喷洒杀虫剂。每次移动的时候，Old MacDonald会在上下左右四个方向中选择一个并移动整数米。当他移动的时候，他会在走过的每个格子上喷洒杀虫剂当Old MacDonald停止喷洒杀虫剂后，他将他所有的移动记录在一张纸上。现在，他想知道，有多少小块不会受到科罗拉多甲虫入侵的影响。一开始，一些土地边缘的小块被入侵。接着，任何没有撒过杀虫剂且有一个相邻小块被感染的尚未被感染的土地也会被感染。帮助Old MacDonald计算有多少小块不会受到科罗拉多甲虫的感染。 | 模拟+暴力 | 时间O(n^2)  空间O(n^2) |
| 8761 | Monsters and Diamonds | Piegirl 发现一只怪物和一本关于怪物和馅饼的书。当她在读这本书的时候，她发现有 n 种怪物，每种都有一个唯一的 1 到 n 的编号。如果你喂怪物一块饼，它就会分成一定数量的怪物，以及至少一个多彩的钻石。怪物们可能存在多种分裂方式。最开始 Piegirl 有且只有一只怪物。她先喂了一块饼，它随之分裂。对于分出来的怪物，继续喂饼，直到它们都分裂为钻石。然后她把所有钻石收集起来。你将得到一系列规则描述不同怪物的分裂方式，每种怪物至少有一种分裂方式。分裂的时候，如果有多种方式，Piegirl 可以选择任意一种。你的任务是：对于每种怪物，确定以其为起始，Piegirl 可以得到的钻石最少最多分别是多少。Piegirl 有无限多的饼。 | 最短路 | 时间O(mlgn)  空间O(m) |
| 9165 | Photo | 有N头奶牛排成一排，标号为1..N。小明拍了M张照片，照片i包含了从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。 | 动态规划 | 时间O(n)  空间O(n) |
| 9166 | Optimize! | http://www.tsinsen.com/RequireFile.do?fid=AGNdAH8e  优化这段代码。 | 线段树 | 时间O(nlgn)  空间O(n) |
| 8757 | Tourists | 位于Ultima Thule的公园内的一条双重观光道路是由以下原理工作：  \*我们引入直角坐标系。  \*在一些时刻会有2个游客同时从点（-1,0）和（1,0）出发（去散步）。其中第一个人出发点（-1,0）,第二个人出发点为（1,0）。  \*每一对游客都以相同的速度1行动（每秒前进单位长度）。第一个人沿直线x=-1、第二个人沿直线x=1行动。他们都向y轴正方向移动。  \*在一些时刻墙会出现。墙（li,ri）是一条在点（0,li）和（0,ri）之间的线段。每个墙都瞬间出现。Ultima Thule官方想要了解对于每一对同时出发的游客，他们将会有多长时间无法彼此望见？2个游客不能看到对方当且仅当他们位置的连线与至少1面墙相交。2条线段相交是指他们至少有1个公共点。我们假定线段的端点属于这条线段。帮助他们计算所要求的时间。 | 平衡树 | 时间O(nlgn)  空间O(n) |
| 9203 | Dima and Figure | Dima喜欢在一块长方形纸片上作自己喜欢的画。  一块大小为N\*M的长方形纸片，包含着N行与M列，并且初始所有格子都是白色的。  Dima每一次作画，会把纸片上一些格子涂黑，并定义这幅“画”就是所涂黑的格子。  Dima喜欢一幅画仅当：  \*包含至少一个涂黑的格子。  \*所有的涂黑的格子形成一个连通块，换句话说，你可以从任意一个涂黑的格子移动到另一个任意涂黑的格子（一个格子可以移动到四联通的格子里）。  \*从一个涂黑的格子(x1,y1)到另一个涂黑的格子(x2,y2)所需的最少移动步数等于|x1-x2|+|y1-y2|。在N\*M的长方形纸片上，他能画出多少幅自己喜欢的画呢？ | 动态规划 | 时间O(n^2m)  空间O(n^2) |
| 8741 | The Last Hole! | 在平面上有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。我们可以想象成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。我们定义一个白色的封闭区域为一个’洞 ’，询问最后一个洞消失的时刻。 | 计算几何 | 时间O(n^3)  空间O(n) |
| 8742 | Cow Neighborhoods | 了解奶牛们的人都知道，奶牛喜欢成群结队。观察约翰的N只奶牛，你会发现她们已经结成了几个“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 | x1 - x2 | + | y1 - y2 | <= 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。给出奶牛们的位置，请计算有多少个牛群，以及最大的牛群里有多少只奶牛。 | 并查集 | 时间O(nlgn)  空间O(n) |
| 9157 | Shaass and Painter Robot | 给你一个N\*M的网格，一开始都是白色的。上面有一个机器人，一开始位于格子（X，Y）上，面朝某个方向（左上，左下，右上，右下之一）。然后机器人会一直顺着这个方向走下去，每当遇到边界时会遵循光的反射定律改变方向，然后继续走。每当机器人走到一个格子后，它会将这个格子染黑，用掉一个单位颜料。即便这个格子已经被染黑了，也需要耗费一个单位颜料。当机器人意识到这个N\*M的网格已经变成黑白相间的时候，它会立即停止行动。现在希望你求出，机器人停下来的时候，已经耗费了多少颜料？或者指出永远不可能停下来。 | 暴力 | 时间O(n)  空间O(n) |
| 9222 | Doodle Jump | 在Doodle Jump中，你要控制一只四条腿的动物，名为“Doodler”，在一个永无休止的平台上，一直跳高而不堕下，便能获得高分。  Doddle Jump是一款非常流行的游戏，小岛也非常喜欢玩它。有一天在玩这个游戏的时候，她在想是否存在一个因为Doodler的跳跃能力不够而跳不上的平台。考虑以下的问题：有n个平台。第x(1≤x≤n)个平台的高度是a·x mod p，其中a和p是互质的正整数。Doodler最大可能跳的高度是h。也就是说，它可以从高度h1跳到h2(h1<h2)，如果h2-h1<h的话。一开始Doodler站在高度为0的地上。现在的问题是它能不能跳上最高的平台。 | 数学 | 时间O(lgn)  空间O(1) |
| 9212 | Theft of Blueprints | 给出一个n个点的带权无向图，满足对于任意一个大小为k的顶点集合S，恰好有一个点与S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。现在求对于一个大小为k的子集操作代价的期望。 | 数学期望 | 时间O(n^2)  空间O(n^2) |
| 9214 | Olya and Graph | Olya现在有一张n个点、m条边的有向无权图。现在我们以某种方式给所有点从1到n标号，从而保证原图中任意从u到v的有向边满足不等式：u<v。现在Olya想知道有多少种方案添加任意数量（可能是0）的有向边，使得该图满足下列条件：1、 从点i出发，可以到达点i+1，i+2,…..,n。2、 任意从u到v的有向边满足不等式：u<v。3、 两点之间最多有一条边。4、 对于一对点i、j(i<j)，若j-i<=k，那么从i到j的最短距离等于j-i条边。5、 对于一对点i、j(i<j)，若j-i>k，那么从i到j的最短距离等于j-i或j-i-k条边。 | 动态规划 | 时间O(n)  空间O(n) |
| 9224 | The Red Button | Piegirl终于发现了红色按钮，你现在还剩最后一个机会去改变这个结局。这个按钮下面的电路由n个从0到n-1编号节点组成。为了关闭这个按钮，这n个节点必须以特定的序列拆解。节点0必须首先拆解，在拆解了节点i后，下一个被拆解的节点必须是(2·i) mod n或(2·i)+1 mod n。最后一个被拆解的节点必须是节点0。节点0必须被拆解两次，其他节点必须刚好被拆解一次。你的任务是找到一个符合要求的顺序并输出它。 | 乱搞 | 时间O(n)  空间O(n) |
| 9179 | Greedy Elevator | CodeForces总部有一幢m层的办公楼，其中有一个先进的电梯控制系统，工作方式如下：所有办公楼层用1-m依次编号。在t=0的时刻，电梯位于第1层。电梯是空的而且别的楼层上没有人在等待。接着，在时刻ti（ti>0）会有人来到某楼层等待电梯。为了简便，我们假设一个人在我们记录下的时间段里只用了一次电梯。对于每一个人我们知道三个参数：他开始等电梯的时刻，他初始位于哪个楼层以及他想去哪个楼层。电梯的移动方式如下：在一个时刻t（t>=0）电梯肯定停在某一个楼层。电梯中所有想去这个楼层的人会出电梯，然后所有在这个楼层等待的人会进入电梯。如果有一个人刚好在时刻t来到该楼层，那么他也有足够的时间进入电梯。我们可以假设以上的过程都是瞬间完成的。然后，电梯需要决定向哪个方向开动。电梯决定的方式如下：  （1）如果电梯已经是空的而且整幢楼中都没有人在等电梯，那么电梯在t+1时刻仍然会保持在那一楼层  （2）否则，如果电梯在楼层x，我们定义p\_up表示电梯中要到编号比x大的楼层的人和当前时刻t在编号比x大的楼层等电梯的人的总数，p\_down表示电梯中要到编号比x小的楼层的人和当前时刻t在编号比x小的楼层等电梯的人的总数。如果p\_up>=p\_down，那么电梯在时刻t+1会到楼层x+1，否则电梯会到楼层x-1。你的任务是模拟电梯的工作并且告诉每一个人他能在哪个时刻到达他想去的楼层。 | 平衡树 | 时间O(nlg^2n)  空间O(n) |
| 8772 | Meeting Her | Urpal住在一个大城市。这个城市有n个交叉点编号从1到n。交叉点被m条有向的街道连接，所有的道路有相等的长度。Urapl住在交叉点a，约会被安排在一个在交叉点b的餐馆中。他想乘坐公交去交叉点b。这里有k个公交车公司。在每一秒的开始，一个公交车从第i个公司随机选择一条从交叉点si到交叉点ti的最短路径然后通过这条路径。可能没有从交叉点si到交叉点ti的路径。在这种情况下没有公交车会离开si去ti。如果一个公交车通过Urpal所在的交叉点，则Urpal可以上这辆公交车。他可以在中途任意一个交叉点下车。现在Urpal想知道是否有可能乘坐公共交通在有限的时间内到达约会地点，和最坏情况下他需要乘坐公交车的次数。在任何时刻Urpal只知道他自己的位置和约会地点。当他上了公交车时他只知道这辆公交车属于第几个公司。当然Urpal知道城市地图和每个公司的(si,ti)。 | 动态规划+最短路 | 时间O(n^3)  空间O(n^3) |
| 9185 | Cows in a Skyscrape | 在奶牛爬到了他们最喜欢的那座摩天大厦的顶部后，她们有了一个问题。奶牛拒绝使爬楼梯回去，她想用电梯回到1楼。电梯有W的磅的最大承重，第i只奶牛有Ci 磅的体重。请你帮助贝茜算出最少坐多少次电梯才能使所有的奶牛回到第1层楼。每个电梯里的奶牛的总体重不能超过电梯的最大承重。 | 搜索 | 时间O(?)  空间O(?) |
| 9162 | Ksusha and Square | 这天Ksusha碰到一个面积不为0的凸多边形。她现在想知道：如果她随机选择两个不同的格点(格点可以在凸多边形内部或者边界上面)，并以它们之间的连线为对角线做一个正方形，那么这个正方形的面积期望是多少？其中两个格点p,q(p!=q)与q,p被视作相同。帮助Ksusha计算出所需的期望面积！ | 数学期望 | 时间O(nlgn)  空间O(n) |
| 8775 | The Evil Temple and the Moving Rocks | 你现在正在一个邪恶神庙里进行探险旅行。在战胜了两只小怪后，你到达了一个n\*n的正方形房间中，四周都是围墙。在房间的尽头有一扇被邪恶的魔法力量锁住的门。房门上刻着这样一段话：石块激烈碰撞的声音将会唤醒这扇大门！  作为一名高级探险家，你立刻明白了其中所蕴含的奥秘。与此同时，相邻的一个房间里有无数块具有魔力的石头。这些石头都属于下列四种类型之一：  • ’^’：这种类型的石块将会向上移动；  • ’<’：这种类型的石块将会向左移动；  • ’>’：这种类型的石块将会向右移动；  • ’v’：这种类型的石块将会向下移动；  要想打开这扇门，你首先需要在一些格子中放上某种类型的石块。然后，你可以选择之前放置的其中一块石头并激活它。被激活的石块将会一直朝着它的方向移动，直到撞到了其他石块或者撞到了房间四周的围墙（如果在它的方向上紧挨着就有其他石块，它将不会有任何移动）。之后这块石头将停止运动。如果它撞到了围墙，则游戏结束。否则，它所撞击到的石块将被激活，且这一过程将会持续发生。但是，当所有石块被激活的总次数达到107后，即使游戏还在进行，游戏也将被强行停止。倘若石块在撞击到围墙或者其他石块之前至少移动了一步，则该次撞击将会发出一记清脆的响声。当石块撞击发出的响声达到或者超过x次，房间的门将会被打开。在这之后，石块的移动仍将继续。 | 构造 | 时间O(n^2)  空间O(1) |
| 9225 | Sheep | 在一个大农场有一片草地和成群的羊。这里总共有 n 只羊，它们中的每一个被标上了一个独一无二的编号，编号的范围从 1 到 n。这片草地有无限数量的区域，从 1 到无穷编号。已知 i 号羊喜欢编号从 li 到 ri 的区域。这里有两个牧羊人在照顾羊群：甲和乙。甲每天早晨起得很早，领着羊群到草地上去吃草。乙则在晚上出来领回所有的羊。一天早晨，甲没有时间领羊群到草地上吃草。因此，他把所有有公共喜欢区域的羊拴在一起。甲认为这样会更好——乙晚上时会少做些工作，因为羊不会走开太远，它们正被互相拴着呢！到了夜晚，乙来到了草地，他把羊群聚集起来并尝试把它们排成一队。但是尽管他很努力了，羊群还是没有排成他想要的那样。乙没有力气和能力去解开羊群，只好让它们保持原样，但有一个条件：他想要把羊群排成一队，使得相互栓着的羊的最大距离尽可能小。两只羊的距离定义为它们在队列中排名的差值。 | 二分+贪心 | 时间O(n^2lgn)  空间O(n^2) |
| 8782 | Roadside Trees | Liss是一只喜欢吃坚果的松鼠。它要求你种一些坚果树。在一条直线有n个位置可以种树，自西向东标号1~n。每个月，每棵树都会长高1米。每个月初，你都会收到一个要求。要求有两种类型：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。在做完这个要求后，你需要求出当前树高的最长上升子序列。一个树的序列，当从西向东的每棵树都严格高于序列中位于它西边的树。序列的长度就是序列中树的个数。特别的，Liss不喜欢高度相同的树，所以任何时刻树的高度是不同的。 | 平衡树+动态规划 | 时间O(nlgn)  空间O(n) |
| 9182 | Mystic Carvings | 北极熊们发现了一块带有神秘雕刻的巨大圆形浮冰。浮冰上总共有n条雕刻的路径，每条路径连接两个在浮冰边缘的节点,这些endpoints被逆时针编号为1,2,…,2n。每个endpoint都被且仅被一条路径连接。现在有6只北极熊决定在endpoints上建造他们自己的洞穴，每只北极熊的洞穴是独立的。Alice和Bob是一对基友，他们相信浮冰上雕刻的路径是外星人的杰作，有着神秘的力量，因此他们决定把他俩的洞穴建造在某条路径连接的两个endpoint上。Carol和Dave，Eve和Frank也是基友，所以他们也决定这么做。我们定义洞穴X和洞穴Y的距离为：从洞穴X走到洞穴Y至少需要经过的其他洞穴数+1。为保证公平，每对基佬的洞穴的距离必须都相等。现在问题来了:给定N和N条雕刻的路径，求三对基佬有多少种公平的建造洞穴的方案。 | 线段树 | 时间O(nlgn)  空间O(n) |
| 9170 | Endless Matrix | 有一个矩阵，里面有连续的数字，从1开始。 若a\_{i, j} < a\_{t, k} (i, j, t, k ≥ 1) 当且仅当  1. max(i, j) < max(t, k)  2. max(i, j) = max(t, k) 且 j < k  3. max(i, j) = max(t, k), j = k 且 i > t  给定 x\_1, y\_1, x\_2 和 y\_2 (x\_1 ≤ x\_2, y\_1 ≤ y\_2) 求http://espresso.codeforces.com/94c30c938cfe895f10e965c38eaded3f1a3707c0.png的后10位。 | 数学 | 时间O(1)  空间O(1) |
| 9164 | Cyclical Quest | 询问“有多少s的连续子串是周期性同构于一个给定的字符串x”。您将得到的字符串S和N个字符串Xi，对于每个字符串Xi，有多少s的连续子串是周期性同构于这个给定的字符串Xi？ | 后缀数组 | 时间O(nlgn)  空间O(nlgn) |
| 9180 | Summer Homework | 聪明的海狸掌握了所有算数运算方法，于是开始完成老师布置的令人瞠目结舌的暑期作业。  　　初始有一个整数序列a1, a2, …, an。你的任务是对它进行m次操作，有以下3种：  　　1. 输入x和v，将ax赋值为v  　　2. 输入l和r，你需要计算这样一个和：http://www.tsinsen.com/RequireFile.do?fid=J2M3bt3N  　　其中f0=f1=1，且i>=2时，http://www.tsinsen.com/RequireFile.do?fid=bTrD8LJd  　　3. 输入l, r和d，你需要对al到ar区间加上数d | 线段树 | 时间O(nlgn)  空间O(n) |
| 9192 | Transferring Pyramid | Vasya和Petya开始用起了一种新的数据结构:金字塔。一个金字塔有n行，第i行有i个格子，每一行都相对于上一行左移了半格。金字塔的格子依次被标号为1,2,...,n(n+1)/2。这个数据结构支持两种修改操作：  1.单点修改，即指定一个格子修改成特定的值，需要传入三个参数："t i v",其中t=1(表示是1类操作，i表示修改格子的编号，v表示修改成的值2.子金字塔修改，图中高亮的部分就是一个以5号格子为顶端的子金字塔。对于这种修改，需要传入s+2个参数:"t i v1 v2 ... vs"，其中t=2，表示是2类操作，i表示子金字塔的顶部格子编号，s是这片子金字塔的大小，vj表示子金字塔的第j个格子应该传入的值。Vasya和Petya有两个金字塔，每人一个，Vasya改变了他的金字塔中的一些格子，他现在想把他执行的操作序列发给Petya，让他来重复自己的操作。在所有可能的操作序列中，Vasya想找到最短的(包含最少数字个数的)，来减少发送成本。而找到最短序列的任务就交给你了。你有一个n行的金字塔，有k个格子需要被修改，找到一个最短的修改序列，使得这k个格子中的每一个都至少被一个修改操作覆盖。 | 动态规划 | 时间O(nsqrtn)  空间O(n) |
| 9161 | Binary Key | 假设有p和q两个长度为正整数的字符串，我们分别叫它们为匣子和钥匙。其中，钥匙串q只包含字符0和1。  下面给出一段简单的算法用以描述如何从匣子串p中提取出信息串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;  }  在上面一段伪代码中 i,j是整形变量，s是字符串， "="是赋值运算符, "=="是一个比较运算符,"[]"是获取字符串某个下标上的字符的运算符,"<>"是一个空字符串。我们假定所有字符串都是从下标0开始的。  你需要构造一个字典序最小的长度为k的钥匙串。使该钥匙串用于以上算法时，从给定的匣子串p中获得给定的信息串s | 动态规划+贪心 | 时间O(n)  空间O(n) |
| 9215 | Pilgrims | 在很久以前有一片土地被称为Dudeland。Dudeland包含n个城镇，它们用n-1条双向道路连接起来。这些城镇通过道路可以两两互达。这里有m个修道院坐落在m个不同的城镇。每个修道院有一个教徒。  在一年之始，每个教徒会选择离他最远的一个修道院。如果有多个，他会把所有的都列入清单。在“Big Lebowski day”里，每个教徒会随机选择一个清单里的城镇开始走去。  Walter讨厌教徒。他想尽可能的通过阻止他们的行程来让尽可能多的人不开心。他计划摧毁一个没有修道院的城镇。一个教徒如果在他的清单里没有任何一个城镇能去，他就会不开心。你需要求出Walter最多能让几个教徒不开心。除此之外，你还要计算他有多少种方法。 | 动态规划 | 时间O(nlgn)  空间O(nlgn) |
| 9194 | Polo the Penguin and Lucky Numbers | 定义幸运数字是正整数,同时它们的十进制数表示只能包含幸运数字4和7。企鹅Polo有两个正整数l和r(l < r),他们都是幸运数字。此外,他们的长度(数字的十进制数表示，没有前导零)是相等的。假设在l与r之间（包括l和r）有n个不同的幸运数字，ai表示序列里的第i个(以升序排列)。请计算出a1·a2 + a2·a3 + ... + an - 1·an。 | 数学+动态规划 | 时间O(n)  空间O(n) |
| 9198 | Rectangle And Square | 你有N个矩形。所有矩形的四个角都是整点，并且两组对边分别平行于X和Y两坐标轴。不同的矩形可能接触，但是不会重叠。现在你的任务是：判断是否存在一个构成正方形的子集。 | 动态规划 | 时间O(nsqrtn)  空间O(n) |
| 9204 | Race | TOC是一个长方形的城市，有着N\*M个方格状的街区。若干建筑物，双向直线的道路以及交叉路口组成了TOC。每个建筑物：占恰好的一个街区。每条道路：所有道路的宽度都为1个街区，并且道路都是水平的或竖直的。每个交叉路口：占一个街区，位于道路的交汇处。  我们称两个街区是相邻的，当且仅当它们之间存在着公共边。没有两条道路或两个交叉路口是相邻的。在每年的狂欢节中，TOP总会按照一条特殊的路线游行。TOP会从一个街区出发，接下来经过若干个交叉路口，最终在一个街区停下。TOP知道从某一个街区到其相邻的街区所花费的时间。同时，从交叉路口需要花费1分钟来到达相邻的街区。TOP不能经过有建筑物的地方。我们知道TOP的初始与结束位置，也知道TOP途中经过的交叉路口的顺序。但他完成游行之后，他会一直呆在结束位置。你的问题是，找出在他出发了K分钟之后，位于哪个位置。TOP总会沿着最短的道路经过给定的交叉路口，而到达终点。 | 字符串 | 时间O(n^3)  空间O(n^2) |
| 9220 | Ladies' Shop | 一位女士最近在“天涯海角”这座城市新开了一家店。为了准备开业，这家店买了n个包，每个包可以承受ai的重量。奇怪的是，你不能用这些包装总重量严格小于ai的物体。现在，这家店里还没有规定好要卖的的东西的重量。而这时候就需要你来决定这些东西的重量。你的任务是找到这样一个集合p1, p2, ..., pk (1 ≤ p1 < p2 < ... < pk) 代表物体的重量，并要满足以下条件：1. 每一个都必须被使用。也就是说，对于第i个背包，必然会有这样一堆物体，它们的重量的总和等于ai。我们可以认为每种物体都有无限个，你可以在同一个背包里放入若干个相同重量的物体。2. 对于任意一种选取方案，如果这些物体总重量小于等于m，那么必须有一个背包可以放下这些物体。类似的，你可以在一种方案里选择若干个相同重量的物体。3. 在满足条件一和条件二的同时，k必须尽可能的小。寻找并打印出合法序列。 | Fft | 时间O(nlgn)  空间O(n) |
| 9201 | Context Advertising | 在这个问题中我们会看一些广告环境的简单版本。你得到了一个包含n个单词的文本。一个正式的广告标语有恰好r行，每一行最多包含c个字符。潜在的顾客总是喜欢看到很多广告，因此你应该确定文本中能写在标语上的最长的连续单词。标语中一行中的单个单词应该用空格隔开。你可以一次性插入多于一个空格。你不能把单词断开，就是说，每一个单词必须完整的在标语中的某一行中出现。另外，你不能改变单词的顺序。单词数应当越多越好。输出你所做的标语 | 倍增 | 时间O(nlgn)  空间O(nlgn) |
| 9159 | Torcoder | 给出一个长度为N的字符串，以及M个操作。每个操作给出一个区间[l,r]，你需要将[l,r]中的字符重排，使得这个子串成为一个回文串——如果有多种方案，你要使得字典序最小。每个操作结束后都不撤销，也就是将字符串依次进行M次变换。如果某个操作不可能排出回文串，则直接无视这个操作。你需要输出经过M次操作后的字符串。 | 线段树 | 时间O(nlgn)  空间O(n) |
| 8785 | Deja Vu | 众所周知，我们生活在黑客帝国的母体里很久了。现在由海狸来统治第七母体。尼尔是一只海狸。当尼尔能看到他即将去的或者已经去过的地方所发生的事情，他会说"WOW"。具体地：尼尔的城市可以用一个n个点m条边的有向图来描述，没有两条边连接了同一对点。尼尔经过一条边的时候，尼尔会看到一些点的编号。无论他经过同一条边多少次，这些编号都是同样的、并按照同样顺序给出的。一条路径能让尼尔说"WOW"，当且仅当，这条路径上的点构成的序列，和这条路径上的边上的编号构成的序列完全一样。要求统计长度为1,2,...,2\*n的能让尼尔说"WOW"的路径的个数。 | 动态规划 | 时间O(n^3)  空间O(n^3) |
| 8778 | Maximum Waterfall | Emuskald 被雇佣建造一座符合最近园林建筑学趋势的人造瀑布。一个现代的人造瀑布包括许多个嵌在墙壁中的水平板，水从一块水平板流向另一块，直到它从墙的顶端流到底端为止。墙的高度是t，有n块水平板嵌在上面。每块水平板可以看作一条水平的线段，高度为hi，从li到ri。第i块水平板连接的平面上的(li,hi)与(ri,hi)两点。墙的顶端可以看作一块在(-10^9,t)到(10^9,t)间的水平板。相似的，墙的底端可以看作一块在(-10^9,0)到(10^9,0)间的水平板。不会有两块水平板之间有公共点。Emuskald 知道，为了瀑布看上去漂亮，水可以从第i块板流向第j块板当且仅当：1. max(li,lj)<min(ri,rj)（二者的水平位置存在相交部分）2. hj<hi（第j块板在第i块板下方）3. 不存在k(hj<hk<hi)，且(i,k)与(k,j)均满足以上两个条件。从i到j的流量为min(ri,rj)-max(li,lj)，相当于二者的水平相交部分的长度。Emuskald 决定在他的瀑布中，水会沿着一条单向的路径从顶部流到底部。如果水流到了一块水平板上（除了墙的底部），水会流向恰好一个更低的水平板。整个瀑布的水流量被定义为水流路径上每连续的两块水平板间的水平相交部分长度的最小值。  为了建造一座真正好的瀑布，Emuskald 必须要最大化瀑布的流量。请帮助Emuskald维护他的名誉并找到可能的最大水流。 | 平衡树 | 时间O(nlgn)  空间O(n) |
| 9197 | Road Repairs | Berland国有N个城市，编号为1的是首都。有些城市被它们之间的单向的道路连接。然而，并非所有的道路都工作良好。对于每条道路，都知道它的状况（是否需要修理），在需要修理的道路被修理之前，人们无法使用它。Berland政府可以修理这样的道路，使得它们可以重新被使用。隔壁的国家打进来了！为了保卫领土，Berland政府要往每个城市派遣一支军队。然而军队也只能沿着工作良好的道路移动，并且由于军情紧急，已经没有时间修建新的道路。这就是说，为了到达某些城市，可能需要修理若干条道路。同时，为了把尽量多的资源花费到军备上，Berland政府希望修理尽量少的道路，使得从首都出发，通过工作良好的道路和修理过的道路，可以到达所有城市。你需要计算这个最小修理道路数，并指出需要修理的是哪些道路。 | 图论+贪心 | 时间O(m)  空间O(m) |
| 9205 | Biologist | SmallR是个生物学家，她最近的一次研究就是关于怎样改变狗狗的性别，换句话说，她可以把雄性狗变成雌性狗，反之亦然。她正准备展示她的成果。现在她拥有n条狗，改变每条狗的花费不一定相同，这n条狗分别被编号1-n，改变编号为i的狗的代价为vi元，每条狗只能被改变一次。这个实验引起了社会的广泛关注，有m个富人对实验很怀疑 ，他们都想要强行和SmallR打赌，如果SmallR赢了，那么第i个富人会给SmallR Wi人民币，但是奇怪的是他们有特殊的方法来判定是否是SmallR赢了。对于第i个富人，提前地，他将指定确定的ki只狗狗以及他希望狗狗所变成的性别，只有在某一天那些指定的狗全部变为指定的性别这个富人才会认为SmallR赢了。如果SmallR不能使某些不是她朋友的人满意的话她不用支付金钱给那个人，但是如果她不能使她的好朋友满意的话她必须支付给好朋友g人民币作为她对好朋友的失误的道歉。SmallR希望从这个实验里获得尽可能多地金钱，请算出SmallR能得到的最多的金钱是多少 | 网络流 | 时间O(n^4)  空间O(n^2) |
| 9196 | Cow Run | 农民约翰和贝茜给奶牛们发明了一种新的运动游戏。奶牛们在长度为M(2 < = M < = 1000000000)的环形轨道上从相同的位置开始跑步。这个游戏会进行N(1<=N<=14)回合，需要使用8 N张卡片,每张卡片上面都写有一个数字X\_i(0 < = X\_i < M)。每回合约翰取出最前面的8张卡片，留下前4张或者后4张，贝茜继续从约翰选出来的4张卡片中留下前2张或者后2张。接着约翰就会让奶牛们跑R\*X\_top的距离，R表示奶牛们已经跑过的总距离，X\_top表示贝茜留下的2张中的第一张，然后贝茜会让奶牛们跑X\_bottom的距离，X\_bottom表示贝茜留下的2张卡片中的第二张。约翰担心运动后，如果奶牛们跑得距离太远，奶牛就会十分疲累而不能回到出发的地方。他认为如果牛最终离他们的起始位置超过K(0 < = K < =向下取整(M / 2))，他们就无法回到家。我们保证如果约翰的选择(留下前4张或者后4张)总是正确的, 无论贝茜怎样选择，他总是能够确保奶牛可以回家。对于每回合,你的任务是确定约翰应该选择哪一半的卡片,来保证不管贝茜如何选择都能够使奶牛回家。贝茜将在输入中提供她的选择，而约翰的选择需要你来指定。 | 乱搞 | 时间O(?)  空间O(n) |
| 9193 | Hill Walk | 这里有n座山。每座山用一条从(x1,y1)到(x2,y2)的线段来描述，x1 < x2，y1 <y2。线段不会相交或接触,包括他们的端点。此外,第一个山满足(x1,y1)=(0,0)。奶牛贝茜从第一座山的(0,0)位置开始。当贝茜在某座山上时，她会一直爬到这座山的尽头，然后从边上跳下来。如果她落在另一座小山上,她就会继续爬那一座山;否则她就会掉到很远的地方直到她安全降落在气垫上面。每座山(x1,y1)—>(x2,y2)视为包含点(x1,y1)但不包含点(x2,y2),所以,当贝茜掉到一个位置上满足x=x1时，就视为落到了那座山上。如果她掉到一个位置上满足x=x2时，就不能视为她落到了那座山上。请计算出贝茜走过的山的总数。 | 平衡树 | 时间O(nlgn)  空间O(n) |
| 8749 | Toys | Bessie的生日快到了, 她希望用D天来庆祝. 奶牛们的注意力不会太集中, 因此Bessie想通过提供玩具的方式来使它们高兴. 她已经计算出了第i天需要的玩具数T\_i. Bessie的幼儿园提供了许多服务给它们的奶牛程序员们, 包括一个每天以Tc (1 <= Tc<= 60)美元卖出商品的玩具店. Bessie想尽可能的节省钱, 但是Farmer John担心没有经过消毒的玩具会带来传染病(玩具店卖出的玩具是经过消毒的).有两种消毒的方式. 第1种方式需要收费C1美元, 需要N1个晚上的时间; 第2种方式需要收费C2美元, 需要N2个晚上的时间(1 <= N1 <= D; 1 <= N2 <= D; 1 <= C1 <= 60;1 <= C2 <= 60). Bessie在party结束之后把她的玩具带去消毒. 如果消毒只需要一天,那么第二天就可以拿到; 如果还需要一天, 那么第三天才可以拿到.作为一个受过教育的奶牛, Bessie已经了解到节约的意义. 帮助她找到提供玩具的最便宜的方法. | 三分+贪心 | 时间O(nlgn)  空间O(n) |
| 9219 | Pumping Stations | 疯狂的科学家Mike申请了一份工作。他的任务是管理一个抽水站系统。这个系统包含n个抽水站，抽水站被标号为1到n。连接两个抽水站的双向管道可以使水在任何一个方向流动。你知道每个管道的宽度——每小时可以流过的最大水量。每个抽水站可以通过这些管道把水从一些站台抽向其他站台，同时在一小时内总的进水量等于总的出水量。Mike的责任是用泵在这些站台之间输送水。遵循上述规则，可以在一小时内将一定数量的水通过管道从站台a输送到站台b。在这期间，其他站台的水不能流入a，同时b站台的水也不能流出去。但是，任意多的水可以流出站台a，或者是流进站台b。如果一小时内有x升水流出站台a，Mike将获得额外的x元工资。根据合同，Mike需要工作n-1天才能得到薪水。在第一天，他选择两个站台v\_1和v\_2，同时在一小时内，他将一定数量的水从v\_1输送到v\_2。然后，在第i天，Mike选择一个之前没有选过的站台v\_i+1，再将一定数量的水在一小内从v\_i输送到v\_i+1。他在第i天输送的水的总量并不取决于他在第i-1天输送的水的总量。Mike需要挣得尽可能多的钱，帮助迈克找到这样的一个排列v\_1，v\_2，……，v\_n使得麦克能够获得尽可能高的薪水。 | 网络流 | 时间O(n^2m)  空间O(n^2) |
| 9229 | Runs | 有一个由小写字母“a”-“z"组成的字符串。每一个极大连续相同的子序列被称为一个“run”。比如说，“bookkeeper”有7个“run”。询问，对于给定的字符串将其重新排列，有多少种不同的排列使得它和原字符串有相同的“run”？ | 动态规划 | 时间O(n^3)  空间O(n^2) |
| 8751 | GCD Table | 有一个n行m列的表格G，G[i][j]=gcd(i,j)。gcd(i,j)是i和j的最大公约数。给你一个正整数数列a[1],a[2],...,a[k]。如果这个序列在G的某行的连续位置出现，则这个序列在表格中出现。询问序列a是否在G中出现。 | 数学 | 时间O(nlgn)  空间O(n) |
| 8752 | k-Maximum Subsequence Sum | 给定整数序列a[1],[2],...,a[n],你应该处理以下两种操作。　第一个操作的输入是“0 i val”，你应该把a[i]赋值为val。第二个操作的输入是“1 l r k”，你应该输出从a[l],a[l+1],...,a[r]取不超过k段的最大不相交子段和。具体地说，你应该选择不超过k对数(x1, y1), (x2, y2), ..., (xt, yt) (l ≤ x1 ≤ y1 < x2 ≤ y2 < ... < xt ≤ yt ≤ r; t ≤ k) ，使得a[x1] + a[x1 + 1] + ... + a[y1] + a[x2] + a[x2 + 1] + ... + a[y2] + ... + a[xt] + a[xt] + 1 + ... + a[yt]的和最大。 | 线段树 | 时间O(nlgnk)  空间O(n) |