

日期	灵茶山艾府选题	难度	标签	思考记录	柯南选题	难度	标签	思考记录	备注
5/18	<a href="#">CF1148E</a> -Earth Wind and Fire  给两个序列 A B, 每次操作从 A 中选择一个数, 减去一部分值加到另一个数上, 求使 A 变成 B 的具体操作。	2300	思维	<p>因为是一个数减, 一个数加, 那么总和是不变的, 所以先判断两个数组的和是不是一样, 然后把a分成两部分, 一部分少的, 一部分多的。然后多的补给少的就完事了。</p> <p>但题目有限制, 互补的时候少的那个数量一定要比多的那个数量小。这是我一开始没有想到的, 所以wa了好多发。</p> <p>然后我想到的是把少的那部分按照个数排序, 多的那部分按照个数排序, 然后取两部分最小的进行检查并且互换。但这样也是不对的。因为会导致多的那部分的最小值变得很小, 导致它后续不能和其他的进行互换。所以得取少的那部分比它小的最大的那个和他换。</p> <p>Ox3F 补充: 我一开始想的是用最大的补最小的, 但是后来构造出了个反例: 1 3 9 11 2 2 10 10 这样 1 和 11 补完之后, 3 和 9 就没法补了。 后来想到补完之后相对顺序是不变的。 这样可以用栈来维护。</p>	<a href="#">1379D</a> -New Passenger Trams  环形区间覆盖最多/最少点问题	2300	<a href="#">扫描线</a> 排序+双指针	<p>1. 一开始想的是求补集, 转换成最多覆盖, 前缀和之类的; 2. 然后发现「货车被覆盖」可以转换成一段区间, 就转换成了扫描线问题; 3. 求扫描线上最小的货车个数就是答案。</p> <p>其实我想复杂了, 本质上是一个区间覆盖点的问题。 但从另一个角度看, 上面的算法本质上就是一种双指针。这个模型可能对于某些条件更容易思考。</p>	
5/19	<a href="#">CF610D</a> -Vika and Segments  若干宽为 1 的矩形的面积并  <a href="#">本质是求若干线段的交点个数</a>	2300	扫描线	<p>思路很显然, 首先根据x坐标进行排序, 然后维护当前时刻所覆盖的y坐标的区间长度, 然后以两个点x坐标的差乘当前总区间长度就是这个区间的贡献。那么需要一个区间修改, 区间查的一个数据结构, 也就是线段树。思考的时候一开始就想到了它的本质其实就是求长方形面积, 也就是线段树的一个经典题型。写的时候线段树的每个点的状态没有想清楚, 它应该表示的是<b>该点到下一个点之间的距离</b>, 但是我写的维护的是这个线段对应的坐标之间的距离。</p> <p>假如现在有4个点 1 10 13 20, 那么线段[1,2]应该表示的是1到13的距离, 也就是13 - 1 = 12.但是我表示的是10 - 1 = 9.那么这个时候我的递推则会有问题, 如[1,4] = [1, 2] + [3, 4] = 10 - 1 + 20 - 13 = 16.但是正确答案应该是[1, 4] = [1, 2] + [3, 4] = 13 - 1 + 7 = 19.</p> <p>所以这个题的正确思路应该是首先把块的坐标转换成线坐标, 然后套板子就完事了。写的时候要注意两个地方,</p> <pre>for (int i = 1; i &lt;= ncnt; i++) {     ans += 1LL * res[1] * (1LL * nodes[i].x - ltx);     // debug(nodes[i].x, nodes[i].y1, nodes[i].y2, nodes[i].flag, ans);      int lidx = lower_bound(yy + 1, yy + 1 + ycnt, nodes[i].y1) - yy;     int ridx = lower_bound(yy + 1, yy + 1 + ycnt, nodes[i].y2) - yy;      ridx--; //右坐标需要减1.     // debug(lidx, ridx);      update(lidx, ridx, nodes[i].flag, 1, ycnt, 1);     // debug(res[1]);     ltx = nodes[i].x; } void push_up(int rt, int l, int r) {     if (cnt[rt] &gt; 0) {         res[rt] = yy[r + 1] - yy[l]; //根据 yy[r+1]递推过来         return;     }     res[rt] = res[rt &lt;&lt; 1] + res[rt &lt;&lt; 1   1]; }</pre>	<a href="#">1342E</a> -Placing Rooks  N*N 棋盘放 N 个车, 要求所有空地都被攻击到, 且有 k 对车互相攻击, 求方案数。	2300	容斥	<p>没想明白, 看题解是一个容斥的过程— . 题解的容斥就是单个 S2(n,k) 的计算公式……</p> <p>另一种思路是把 n 个棋子放入 n-k 列中。不同的列可以看成不同的盒子, 方案数为 A(n,n-k)*S2(n,n-k)。</p>	
5/20	<a href="#">CF429C</a> -Guess the Tree  75/694=10.8%  有一颗不超过 24 个点的树, 除叶节点外的节点都至少有两个孩子。 已知所有子树的大小, 问这样的树是否存在?	2300	搜索+剪枝	<p>看完题后模拟了几个例子, 发现其实就是找对于当前val来说, 看数组中是否存在不少于两个的数的和等于val-1.但我没办法证明对于当前val拿完数之后的结果没有后效性, 即不会影响到后面数的构造。</p>	<a href="#">CF1371E2</a> -Asterism (Hard Version)  给 n 个怪的能力和和一个数 p, 可以战胜能力值不大于你能力值的怪, 每战胜一个怪能力值就加一。 定义 f(x) 为初始能力值为 x 时, 能战胜所有怪物的怪物排列方案数。 求所有的 x, 使 f(x) 不被 p 整除。	2300	组合数学 上下界分析	<p>1. 注意到 f(x) 是单调递增的 2. 在某个范围内, f(x) 都是 good, 范围外都 not good 3. 转换成求上下界 4. 下界根据 <math>x+i&gt;=a[i]</math> 来求, 即打败一个怪前必须有 <math>x+i</math> 个糖果 5. 上界则需要保证, 对于每个 <math>i&gt;=p-1</math> 的 <math>a[i]</math>, 其对应的位置 <math>i-p+1</math> 的糖果个数不能<math>&gt;=a[i]</math>, 否则此刻在 <math>i-p+1</math> 位置上能选到至少 p 个怪物。那么后续肯定存在一个时刻可以选到 p 个怪物。也就是说需要满足 <math>x+i-p+1&lt;a[i]</math></p> <p>注: 对于任意超过上界的 x, 在某个时刻必定会存在可以选择 p 个怪物的情况</p>	

日期	灵茶山艾府选题	难度	标签	思考记录	柯南选题	难度	标签	思考记录	备注
5/21	<a href="#">CF708C-Centroids</a>  202/589=34.3%  给一颗树，你可以断掉最多一条边，链接到另外两个节点上，使新图仍然是一棵树。  问哪些点可以是新树的重心？	2300	重心 换根 DP	学习了一下重心的性质，其中有一个就是一个点的子树的size不超过n/2，那么回到这一题，如果想使u为重心，有两种情况，一种是如果它的子树size都不超过n/2，那么已经满足条件了。第二种情况是有一颗子树的size_a是超过n/2，那么我需要找到这棵子树的子树，它的size_b要小于n/2，并且要尽可能的大，把它拿走之后，这棵子树的size就变成了size_a - size_b,那么判断size_a - size_b 是否小于 n/2 就行了。  其实就是对于每个节点要维护一个东西，即这个节点的子树下，size 不超过 n/2 的最大size是多少就完事了。	<a href="#">CF1007B-Pave the Parallelepiped</a>  112 / 542 = 20.6%  给三个数，从中各选一个因子，求形成的无序 (a,b,c) 元组数	2400	数论 容斥 组合数学	1. 把属于 A B C 的因子分为 7 种 2. 用容斥计算出各自数量 3. 枚举选这 7 种的不同组合情况，每种合法情况计算组合数	
5/22	<a href="#">CF995C-Leaving the Bar</a>  135/443=30.5%  给 n 个模长不超过 1e6 的向量，每个向量可以取反，要使所有向量之和的模长不超过 1.5e6，哪些向量要取反？	2300	随机化	画了一下发现很难卡掉随机打乱后的数据。 那么随机打乱后贪心。	<a href="#">CF1137D-Cooperative Game</a>  98/470 = 20.8%  交互题：Floyd 判圈，定位起点	2400	Floyd 判圈	根据 Floyd 判圈算法求解。  这题有趣的地方在于操作数的上界的证明。	
5/23	<a href="#">CF901C-Bipartite Segments</a>  138/495=27.9%  给一个只有偶环的无向图，每次查询 [l,r] 有多少个子区间，使子区间内的顶点形成的是一个二分图	2300	仙人掌 DFS 前缀和 二分	在仙人掌上 DFS，求出每个环的最小最大节点，视作一条线段，然后转换成子区间不包含任意线段的问题	<a href="#">CF1442C-Graph Transpositions</a>  291/930=31.2%  有向图，边权为 1，翻转整个图的开销是 2^(k-1)，k 为翻转次数 问 1 到 n 的最短路	2400	01 BFS 分层图最短路	先用 01 BFS 求出到每个点的最少翻转次数 基于此，可以将分层图的空间复杂度优化至 O(20n)	
5/24	<a href="#">CF558E-A Simple Task</a>  给一个字符串，执行 1e5 次操作，每次操作选择一个子串以升序或降序排序，输出最终字符串	2300	线段树	首先思考为什么题会给字符串，而不是整数数组。容易发现字符串其实只有26个数，那么只需要考虑一个区间里26个数的值分别是多少就行了。	<a href="#">CF1004D-Sonya and Matrix</a>  一个矩阵中的 a[i][j] 是根据 (i,j) 到矩阵中的某个点 (x,y) 的曼哈顿距离求出的，给出矩阵中的所有值，求出矩阵宽高以及 x 和 y	2300	想法题	枚举宽高， 根据最后一个符合 4*i 的频数来确定 x 根据最大值来确定 y	
5/25	<a href="#">CF891C-Envy</a>  127/440=28.9%  给定一个带权无向图，每次询问指定若干个边，问这些边能不能全部放到 MST 里	2300	最小生成树 性质	MST 有两个性质 1. 对于任意权值，最小生成树中这种权值的边数是一定的。 2. 对于任意权值，把小于它的权值的边加到MST里后的连通性是一定的  那么在第一次求MST的时候，先根据第2个性质算出每条边在加之前的图的连通性。。然后在每次询问的时候，只需要看边的两点的是否已经联通就行了。	<a href="#">CF1483D-Useful Edges</a>  251/891=28.2%  给定一个图，以及q个tuple(u,v,w),对于一个tuple来说，从u到v经过的edge的边权和小于等于w，那么经过的边是有用的，问有用的边的个数	2400	最短路，思维	很容易想到如果边(x,y)对于tuple(a,b,l)合法，那么dis(a,x) + eg(x, y) + dis(y, b) <= l  枚举x,y,a,b可以做，但是n^4会超时。  一个巧妙的枚举方式是枚举a,y，那么 dis(a,x) 和 eg(x, y) 都是关于x的方程，dis(y,b) 和 l(a, b) 是关于b的方程。 那么首先枚举b，找到 dis(y, b) - l(a, b) 的最小值mi，然后枚举x，找到dis(a,x) + eg(x, y) + mi <= 0 的方案数。 这样就变成了n^3。	
5/26	<a href="#">CF547C-Mike and Foam</a>  189/1026=18.4%  给定一个数组，有一个集合S，最开始S为空，每次询问给定一个下标idx，如果下标在S里，就把它删掉，如果不在，就把它加进去，每次询问返回集合S里gcd=1的点对数	2300	gcd，容斥， 套路 莫比乌斯函数	求集合里和 x 互质的数的个数=总个数-和 x 不互质的数的个数。  那么转换成求和 x 不互质的数的个数了，这是一个经典的容斥问题。 即：求出与x至少有一个相同的质因子 - 与x至少有二个相同的质因子 + 与x至少有三个相同的质因子。。。  5e5 < 2 3 5 7 11 13 17，即最多有7个质因子，那么枚举(1<<7)个状态就行了。	<a href="#">CF1473E Minimum Path</a>  给定一个图，u到v的最短路表示为备注所示的函数 求1到其余点的最短路	2400	分层图最短路	1. 分成四层，分别表示原图、不算最大值、额外再算一遍最小值、不算最大值且额外再算一遍最小值 2. 从第一层到第四层就是要求的最短路 3. 即便当前的边不是最小或最大，后面转移的时候也会覆盖掉	$\sum_{i=1}^k w_{e_i} - \max_{i=1}^k w_{e_i} + \min_{i=1}^k w_{e_i}$

日期	灵茶山艾府选题	难度	标签	思考记录	柯南选题	难度	标签	思考记录	备注
5/27	<b>CF1276C</b> - Beautiful Rectangle  283/647=43.7%  给 4e5 个数, 请你选出尽可能多的数, 填满一个矩阵, 且矩阵每行每列不能有重复数字	2300	构造	1. 枚举矩形宽, 高<=宽 2. 矩形内元素个数最大值不能超过高 3. 因此需要事先统计元素个数, 求最大宽*高时, 可以按个数从小到大扫, 累加合价值  <pre> for h, s, i := 1, 0, 0; h++ {     s += len(cv) - i     w := s / h     if w &lt; h {         break     }     if h*w &gt; mxH*mxW {         mxH, mxW = h, w     }     for ; i &lt; len(cv) &amp;&amp; cv[i].c &lt;= h; i++ {     } } </pre>	<b>CF1468M</b> Similar Sets  给 n 个集合, 每个集合至少两个元素, 共 2e5 个元素。 问是否有两个集合的交集大小至少为 2?	2300	分块思想 四元环	分块做法: 有两种暴力枚举的做法: 1. 暴力枚举一个set, 然后枚举其他set看是否有两个值一样。这个的复杂度为 O(点总个数*set个数) 2. 枚举点对, 看是否有一组点对属于两个set。这个复杂度为O(点对个数)  当set元素个数>sqrt(m)用第一种, 那么复杂度为O(m*sqrt(m)), 这时候我们考虑的情况有大set和大set的交, 大set和小set的交。唯独没有考虑到的是小set和小set的交。那么小set和小set的交再用第二种方法搞, 那么复杂度同样也不会超过O(m*sqrt(m))  四元环: <a href="https://blog.csdn.net/weixin_43466755/article/details/112985722">https://blog.csdn.net/weixin_43466755/article/details/112985722</a>	
5/28	<b>CF1239C</b> - Queue in the Train  168/545=30.8%  n 个人坐成一列, 每个人都有一个人打算喝水的时间点 ti, 一个人不到点是不会去喝水的。另外, 饮水机在第一个座位的左侧, 也就是说如果有多人同时要喝水, 那么位置靠左的人先起身。一个人在饮水机接水需要时间(但是走路不要时间), 每个人接水消耗的时间一样。由于没人喜欢排队, 所以如果一个人发现在他的前方(编号小于他)的任意一个座位空了(说明那个人正在排队), 他就会在自己的座位上等待, 直到所有在他前方的人都回到座位上。需要注意的一点是, 如果有几个人同时打算起身, 编号最靠前的会先起身, 而靠后的会坐下。	2300	模拟	读错题了, 错误的思路: <pre> for i := 0; i &lt; n    h.Len() &gt; 0; {     if i &lt; n {         if h.Len() == 0 &amp;&amp; cur &lt; a[i].t {             cur = a[i].t         }         for ; i &lt; n &amp;&amp; a[i].t &lt;= cur; i++ {             h.push(a[i].i)         }     }     p := h.pop()     cur += t     ans[p] = cur } </pre> 先将数组按照时间排序, 时间相同的按下标排序。然后用一个队列表示排队中的下标, 和一个堆表示伺机排队的下标: <pre> q, w := []int{}, hp{} for left, i, cur := n, 0, int64(0); left &gt; 0; left-- {     if len(q) == 0 {         if len(w.IntSlice) &gt; 0 {             q = append(q, w.pop()) // 立刻进队         } else {             cur = int64(a[i].t)         }     }     cur += t     for ; i &lt; n &amp;&amp; int64(a[i].t) &lt;= cur; i++ {         if i := a[i].i; len(q) == 0    i &lt; q[len(q)-1] {             q = append(q, i) // 立刻进队         } else {             w.push(i) // 伺机排队         }     }     ans[q[0]] = cur     q = q[1:] } </pre>	<b>CF1408E</b> Avoid Rainbow Cycles  现在有m个set, set里的结点两两相连, 边的权值告诉你了, 另外边有颜色, 的边的颜色等于当前set的下标。如果有一个环上的边的颜色存在多种, 那么是不合法的。求删掉若干个变, 使得当前图是合法的。	2400	建图 最大生成树	建图思路: 对于每个set来说, 建一个虚拟结点, 然后让其他的点都连向这个虚拟结点, 这样表示其他的点能够互相相连。那么建完图之后, 如果存在环的话, 那么就说明是不合法的。此时我们需要删掉一些边使得当前图没有环, 倒着想就是选一些边使得当前图没有环, 要使得删掉的边权和最小, 则需要使选的边的和最大, 也就是最大生成树了。	
5/29	比赛 暂停一天								
5/30	<b>CF825E</b> - Minimal Labels  给一个 DAG, 不保证连通。 需要给每个点标号, 要求每条边的入点标号小于出点标号, 且全体标号是一个 1~n 的排列。 求字典序最小的排列。	2300	拓扑	看上去要选择正序拓扑还是倒序拓扑  造一个反例: 3 1 3 1  所以正序是错的	<b>CF1403B</b> - Spring cleaning  给一颗树, 每次选择2个叶子结点, 进行清扫, 清扫的代价是这两个点最短路经过的边, 叶子节点不能重复选择, 要求最后的方案一定要包括所有的边, 并且代价和最小。 q次询问, 每次询问会在若干个点下加叶子结点, 输出最小代价和。	2300	树链剖分	-	

日期	灵茶山艾府选题	难度	标签	思考记录	柯南选题	难度	标签	思考记录	备注
5/31	<a href="#">CF425D-Sereja and Squares</a>  97/578=16.8%  平面直角坐标系上有 n 个互不相同的点，求有多少个边与坐标轴平行的正方形，其四个顶点均在这 n 个点中。	2300	分块思想	感觉还是分块优化的思想  最坏情况下是所有点集中在一个方形内	<a href="#">CF1394B-Boboniu Walks on Graph</a>  385/1131=34.0%  n个点m条边，每个点的度不超过k (k<=9),每条边有一个权值，所有边的权值是1~m的全排列，现在有一个c数组，ci表示，如果一个点的度数为i，那么从这个点会走到连接这个点的权值第ci小的边，给出所有c的方案，使得从任意一个点走都能回到原处。	2300	图论 集合哈希	因为k小于等于9，容易想到全排列所有的方案，然后check这个方案是否可行。 怎么check呢？实际上从任意一个点往一个方向走能回到原地可以转换成走一步后的点集的还是n。  所以我们需要维护一个走下一步的集合s，这里用集合hash将集合映射成一个值，最后判断这个值是否等于1~n集合的hash的值。  * 集合哈希可以总结一下 * 最好写双hash	