

贪心

李淳风

长郡中学

2025 年 1 月 10 日

前言

贪心，往往非常依赖对题目性质的分析与挖掘，有些时候甚至需要靠一些数学直觉。

所以建议大家在考场上写贪心的时候写个暴力对拍。

当然，贪心还是有一些基本的模型与套路在，弄熟之后或许也能帮上一点忙。

P2587 [ZJOI2008] 泡泡堂

第 XXXX 届 NOI 期间，为了加强各省选手之间的交流，组委会决定组织一场省际电子竞技大赛，每一个省的代表队由 n 名选手组成，比赛的项目是老少咸宜的网络游戏泡泡堂。每一场比赛前，对阵双方的教练向组委会提交一份参赛选手的名单，决定了选手上场的顺序，一经确定，不得修改。比赛中，双方的一号选手，二号选手……， n 号选手捉对厮杀，共进行 n 场比赛。每胜一场比赛得 2 分，平一场得 1 分，输一场不得分。最终将双方的单场得分相加得出总分，总分高的队伍晋级（总分相同抽签决定）。

作为浙江队的领队，你已经在事先将各省所有选手的泡泡堂水平了解的一清二楚，并将其用一个实力值来衡量。为简化问题，我们假定选手在游戏中完全不受任何外界因素干扰，即实力强的选手一定可以战胜实力弱的选手，而两个实力相同的选手一定会战平。由于完全不知道对手会使用何种策略来确定出场顺序，所以所有的队伍都采取了这样一种策略，就是完全随机决定出场顺序。

当然你不想这样不明不白的进行比赛。你想事先了解一下在最好与最坏的情况下，浙江队最终分别能得到多少分。

$$n \leq 100000$$

P2587 [ZJOI2008] 泡泡堂

首先，浙江队得最坏情况就是对手的最好情况，所以只需要写一个函数，求解之后交换顺序再求解一边即可。

如果没有平一场得一分这么一个规定，就是一个田忌赛马模型。可以排序后从大到小考虑我们选手，能赢对手的最强就赢，赢不了就用最弱的消耗掉。

现在多了平局的选项，我们的策略就需要改变一下。首先，如果我方最强能赢敌方最强还是可以直接赢；如果两队最强的选手能打平，我们就考虑最弱的选手：

- 如果最弱的选手能赢一场（赢下对方最弱的选手），那么让他赢下这一场肯定不会差；
- 如果最弱的选手赢不了，那么就让他去消耗掉敌方最强肯定不会变差；

这样就得到了我们的贪心策略。

P2168 [NOI2015] 荷马史诗

一部《荷马史诗》中有 n 种不同的单词，从 1 到 n 进行编号。其中第 i 种单词出现的总次数为 w_i 。Allison 想要用 k 进制串 s_i 来替换第 i 种单词，使得其满足如下要求：

对于任意的 $1 \leq i, j \leq n, i \neq j$ ，都有： s_i 不是 s_j 的前缀。

现在 Allison 想要知道，如何选择 s_i ，才能使替换以后得到的新的《荷马史诗》长度最小。在确保总长度最小的情况下，Allison 还想知道最长的 s_i 的最短长度是多少？

P2168 [NOI2015] 荷马史诗

一部《荷马史诗》中有 n 种不同的单词，从 1 到 n 进行编号。其中第 i 种单词出现的总次数为 w_i 。Allison 想要用 k 进制串 s_i 来替换第 i 种单词，使得其满足如下要求：

对于任意的 $1 \leq i, j \leq n, i \neq j$ ，都有： s_i 不是 s_j 的前缀。

现在 Allison 想要知道，如何选择 s_i ，才能使替换以后得到的新的《荷马史诗》长度最小。在确保总长度最小的情况下，Allison 还想知道最长的 s_i 的最短长度是多少？

k 叉哈夫曼树模板题。普通的 2 叉哈夫曼树就是合并果子，这道题因为是 k 进制的字符串，所以初始时每个单词都是一个节点，每次选出当前出现次数最少的 k 个节点合成一个新节点。注意每次合成会减少 $k-1$ 个节点，所以可能会出现最后一次合成不满 $k-1$ 个节点的情况，需要我们在 $(n-1) \% (k-1) \neq 0$ 时提前补上一些出现次数为 0 的节点，来满足 $(n-1) \% (k-1) = 0$ 。

P1607 [USACO09FEB] Fair Shuttle G

逛逛集市，兑兑奖品，看看节目对农夫约翰来说不算什么，可是他的奶牛们非常缺乏锻炼——如果要逛完一整天的集市，他们一定会筋疲力尽的。所以为了让奶牛们也能愉快地逛集市，约翰准备让奶牛们在集市上以车代步。但是，约翰木有钱，他租来的班车只能在集市上沿直线跑一次，而且只能停靠 $N(1 \leq N \leq 20000)$ 个地点（所有地点都以 1 到 N 之间的一个数字来表示）。现在奶牛们分成 $K(1 \leq K \leq 50000)$ 个小组，第 i 组有 $M_i(1 \leq M_i \leq N)$ 头奶牛，他们希望从 S_i 跑到 $E_i(1 \leq S_i < E_i \leq N)$ 。

由于班车容量有限，可能载不下所有想乘车的奶牛们，此时也允许小组里的一部分奶牛分开乘坐班车。约翰经过调查得知班车的容量是 $C(1 \leq C \leq 100)$ ，请你帮助约翰计划一个尽可能满足更多奶牛愿望的方案。

P1607 [USACO09FEB] Fair Shuttle G

我们发现，只要上了班车，奶牛的起点就已经不重要了，我们就只关心车上每头奶牛要去的终点。如果车上有若干头奶牛要到 y 下车，现在又有若干头终点为 $x(x < y)$ 的奶牛想上车，那么之前不让终点为 y 的奶牛上车，换成终点为 x 的奶牛显然不会更差。
因此我们可以考虑把所有的奶牛按照终点排序，遍历每一批奶牛用线段树维护这批奶牛能有多少头上车即可。

P3634 [APIO2012] 守卫

APIO 王国正被忍者攻击！忍者非常厉害，因为他们在进攻的时候可以躲在阴影里面使得其他人看不到他们。整个王国除了国王居住的 APIO 城堡以外都已经被占领了。在城堡前，有 N 个灌木丛，从 1 到 N 编号，有 K 个忍者躲在恰好 K 个灌木丛后面。APIO 城堡里有 M 个守卫。守卫 i 监视着编号从 A_i 到 B_i 的连续的一段灌木丛。每个守卫都向国王报告在他所监视范围内是否有忍者出现。作为国王的仆人，你需要告诉国王，基于守卫的报告，哪些灌木丛后面一定躲着一个忍者，即对于任何和守卫报告不矛盾的忍者排列方式，在这个灌木丛后面都躲着一个忍者。

你需要写一个程序来输出所有的这些灌木丛的编号。

输入数据保证至少存在一种忍者排列方式满足所有条件。

$1 \leq K \leq N \leq 100000, 0 \leq M < 100000$

P3634 [APIO2012] 守卫

APIO 王国正被忍者攻击！忍者非常厉害，因为他们在进攻的时候可以躲在阴影里面使得其他人看不到他们。整个王国除了国王居住的 APIO 城堡以外都已经被占领了。在城堡前，有 N 个灌木丛，从 1 到 N 编号，有 K 个忍者躲在恰好 K 个灌木丛后面。APIO 城堡里有 M 个守卫。守卫 i 监视着编号从 A_i 到 B_i 的连续的一段灌木丛。每个守卫都向国王报告在他所监视范围内是否有忍者出现。作为国王的仆人，你需要告诉国王，基于守卫的报告，哪些灌木丛后面一定躲着一个忍者，即对于任何和守卫报告不矛盾的忍者排列方式，在这个灌木丛后面都躲着一个忍者。

你需要写一个程序来输出所有的这些灌木丛的编号。

输入数据保证至少存在一种忍者排列方式满足所有条件。

$1 \leq K \leq N \leq 100000, 0 \leq M < 100000$

首先，如果一个区间里面没有忍者，我们可以通过差分给原数组打上标记，这样最后扫一遍就能知道哪些灌木丛里面一定没有忍者；然后我们把可能有忍者的灌木丛重新编号，并得到一些新的区间，就只剩下了一种区间。

P3634 [APIO2012] 守卫

考虑一些特殊情况：如果灌木丛的数量等于忍者的数量，那么所有灌木丛都有人这；如果大区间包含了另一个小区间，大区间可以直接不管；长度为 1 的区间可以直接确定一个位置有忍者。

去掉这些情况之后，我们可以得到左、右端点都递增的若干个区间。这道题其实很像区间覆盖问题：按照右端点排序之后，若当前区间还没有被覆盖，就选择该区间的右端点。这样可以得到一组忍者数量最少的方案。题目要我们求的是必定存在忍者的灌木丛，也就是必定会被选中的位置。一个点是必选点当且仅当不选它的时候无解。由于无解只有一种情况，就是选出的点数量多了（题目保证有解，所有点全选已经判断过了），那么根据贪心策略肯定是选择区间右端点最优，所以我们依次考虑每一个右端点是不是必选点。

根据我们的贪心策略，如果一个右端点 x 没有被选择，那么最优的点就是它的左边一个点 $x-1$ 。我们考虑选择 $x-1$ ，假设总共有 t 条线段， $1 \sim i$ 的线段右端点都小于 $x-1$ ， $j \sim t$ 的线段左端点都大于 $x-1$ ，那么我们只需要找到覆盖线段 $1 \sim i$ 的最少点数 f_i 和覆盖线段 $j \sim t$ 的最小点数 g_j ，如果 $f_i + g_j + 1 > k$ 则说明无解， x 是必选点。

预处理 f, g 数组， i, j 可以用双指针求。

P3243 [HNOI2015] 菜肴制作

知名美食家小 A 被邀请至 ATM 大酒店，为其品评菜肴。ATM 酒店为小 A 准备了 n 道菜肴，酒店按照为菜肴预估的质量从高到低给予 1 到 n 的顺序编号，预估质量最高的菜肴编号为 1。

由于菜肴之间口味搭配的问题，某些菜肴必须在另一些菜肴之前制作，具体的，一共有 m 条形如 i 号菜肴必须先于 j 号菜肴制作的限制，我们将这样的限制简写为 (i, j) 。

现在，酒店希望能求出一个最优的菜肴的制作顺序，使得小 A 能尽量先吃到质量高的菜肴。

也就是说，

- 在满足所有限制的前提下，1 号菜肴尽量优先制作。
- 在满足所有限制，1 号菜肴尽量优先制作的前提下，2 号菜肴尽量优先制作。
- 在满足所有限制，1 号和 2 号菜肴尽量优先的前提下，3 号菜肴尽量优先制作。
- 以此类推。

$n, m \leq 10^5$ 。

P3243 [HNOI2015] 菜肴制作

题目要求的顺序正着来看并不是字典序。但是如果倒着考虑的话，就是相当于字典序最大。

考虑 1 号点，假设有 m 个点拓扑序必然在 1 号点之前，1 号点能出现的最靠前的位置就是第 $m+1$ 个。如果把 1 号点出现的位置后移，只会导致可能的最小倒序字典序变大。

P2048 [NOI2010] 超级钢琴

小 Z 是一个小有名气的钢琴家，最近 C 博士送给了小 Z 一架超级钢琴，小 Z 希望能够用这架钢琴创作出世界上最美妙的音乐。

这架超级钢琴可以弹奏出 n 个音符，编号为 1 至 n 。第 i 个音符的美妙度为 A_i ，其中 A_i 可正可负。

一个“超级和弦”由若干个编号连续的音符组成，包含的音符个数不少于 L 且不多于 R 。我们定义超级和弦的美妙度为其包含的所有音符的美妙度之和。两个超级和弦被认为是相同的，当且仅当这两个超级和弦所包含的音符集合是相同的。

小 Z 决定创作一首由 k 个超级和弦组成的乐曲，为了使得乐曲更加动听，小 Z 要求该乐曲由 k 个不同的超级和弦组成。我们定义一首乐曲的美妙度为其所包含的所有超级和弦的美妙度之和。小 Z 想知道他能够创作出来的乐曲美妙度最大值是多少。

$n, k \leq 500000$

P2048 [NOI2010] 超级钢琴

注意到 k 不大，所以如果我们每次能够快速找出全局最大的“超级和弦”，这个问题就解决了。

由于区间长度为 $L \sim R$ ，所以如果我们确定了区间的左端点为 i 的话，右端点的范围就是 $[i + L - 1, i + R - 1]$ 。由于我们的区间和可以写为两个前缀和的差 $S_j - S_{i-1}$ ($i + L - 1 \leq j \leq i + R - 1$)，在确定了 i 的情况下，想要找到最大的区间和，只需要找到最大的 j 即可，这个我们可以用 ST 表来做到。

这样我们就找出了 $O(n)$ 个固定左端点的最大区间和，自然也就找出了全局的最大区间和。那么我们怎么找次大的区间和呢？考虑把全局最大和代表的区间（假设是 $[l, r]$ ）删掉，对于以 l 为左端点的区间和，只是在可行的右端点范围 $[l + L - 1, l + R - 1]$ 内删去了一个点 r ，范围变成了两个区间 $[l + L - 1, r - 1], [r + 1, l + R - 1]$ ；对于不是以 l 为左端点的区间和没有任何变化。

所以我们将所有固定左端点的最大区间和用堆维护。每次取出堆顶元素，拆成两个再放回堆中。

练习题

P3626

P1954

P2123

P4053

P3545

P3698

P2354

P4437