

贪心乱讲

GAREN 2019.03.27

定义

- ▶ 贪心算法（又称贪婪算法）是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部最优解。
- ▶ 贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关。

From: 百度百科

可分割背包问题

- ▶ 给你若干件物品，物品有体积和价值，求有限容量的最大价值。物品可以分割。
- ▶ 既然可以分割，那就贪心取性价比最高的物品即可。

经典问题——线段覆盖

- ▶ 一条数轴上有 m 条线段，覆盖了 $[s_i, t_i]$ 。
- ▶ 需要你回答三个问题：
 1. 选择尽量少的线段，覆盖 $[0, n]$ 。
 2. 选择尽量多的线段，使得它们不相交。
 3. 选择尽量少的点，使得每条线段上都有点。

From: fstqwq大佬的课件

接水问题

- ▶ 有一群人要接水，每个人接水都有对应的时间 t_i 。求所有人接水的最短时间。
- ▶ 小学生都知道时间短的先排，但是如何证明？
- ▶ 反证法。
- ▶ 如果时间长的跟时间短的换了位置，那么答案会更劣。

最少线段覆盖

- ▶ 预处理：按照左端点或右端点排序。下面的操作都要先排序。
- ▶ 贪心策略：贪心地选左端点在覆盖范围内，右端点尽量大的线段即可。

最多不重叠线段覆盖

- ▶ 每次只需要贪心地取右端点尽量小，左端点不与前面重叠的线段即可。

最少点覆盖

- ▶ 每次都贪心选右端点。

[JSOI2007] 建筑抢修

- ▶ 有若干个建筑要修，每个建筑有价值 and 修理的deadline，超过deadline就再也修不了了。
- ▶ 修理一个建筑需要1个单位时间。求能挽救的最大价值。

[JSOI2007] 建筑抢修

- ▶ 按照deadline排序之后直接贪心？不见得是对的。
- ▶ 我们可以使用反悔的贪心。
- ▶ 如果出现时间不够修的情况，如果花费时间比之前修过的最大时间少，那么可以被替换，答案会更优。
- ▶ 之前的最大时间直接用大顶堆维护即可。

NOIP2012 国王游戏

- ▶ 恰逢 H 国国庆，国王邀请 n 位大臣来玩一个有奖游戏。首先，他让每个大臣在左、右手上面分别写下一个整数，国王自己也在左、右手上各写一个整数。然后，让这 n 位大臣排成一排，国王站在队伍的最前面。排好队后，所有的大臣都会获得国王奖赏的若干金币，每位大臣获得的金币数分别是：排在该大臣前面的所有人的左手上的数的乘积除以他自己右手上的数，然后向下取整得到的结果。
- ▶ 国王不希望某一个大臣获得特别多的奖赏，所以他想请你帮他重新安排一下队伍的顺序，使得获得奖赏最多的大臣，所获奖赏尽可能的少。注意，国王的位置始终在队伍的最前面。

NOIP2012 国王游戏

- ▶ 这道题需要我们用一种证明方法：排序法。
- ▶ 我们在自定义排序函数的时候，只考虑了两个元素之间的比较。
- ▶ 同理，我们在这里也随便挑两个大臣出来比较，通过冒泡证明有序性，然后使用快排来实现这个过程。

NOIP2012 国王游戏

- ▶ 设有两个大臣，手上的整数分别是 a_1, b_1, a_2, b_2 ，国王手上的数是 a_0, b_0 。
- ▶ 如果是0 1 2排的话，答案是 $\max(\frac{a_0}{b_1}, \frac{a_0 \times a_1}{b_2})$ 。
- ▶ 如果是0 2 1排的话，答案是 $\max(\frac{a_0}{b_2}, \frac{a_0 \times a_2}{b_1})$ 。
- ▶ 显然，有 $\frac{a_0}{b_1} < \frac{a_0 \times a_2}{b_1}$ 和 $\frac{a_0}{b_2} < \frac{a_0 \times a_1}{b_2}$ 。
- ▶ 令后者答案更大，则必须有 $\frac{a_0 \times a_1}{b_2} < \frac{a_0 \times a_2}{b_1}$ ，展开得 $a_1 b_1 < a_2 b_2$ 。
- ▶ 所以， $a_i b_i$ 越大的大臣应该排越后，否则答案变大。
- ▶ 更一般性的证明是随意地设两个大臣，把 a_0, b_0 设为前面所有大臣的左右手乘积。讨论交换是否更优。其实是一样的。
- ▶ 最后再套个高精板子就可以过了

NOIP2018 赛道修建

- ▶ C 城将要举办一系列的赛车比赛。在比赛前，需要在城内修建 m 条赛道。
- ▶ C 城一共有 n 个路口，这些路口编号为 $1, 2, \dots, n$ ，有 $n - 1$ 条适合于修建赛道的双向通行的道路，每条道路连接着两个路口。其中，第 i 条道路连接的两个路口编号为 a_i 和 b_i ，该道路的长度为 l_i 。借助这 $n - 1$ 条道路，从任何一个路口出发都能到达其他所有的路口。
- ▶ 一条赛道是一组互不相同的道路 e_1, e_2, \dots, e_k ，满足可以从某个路口出发，依次经过道路 e_1, e_2, \dots, e_k （每条道路经过一次，不允许调头）到达另一个路口。一条赛道的长度等于经过的各道路的长度之和。为保证安全，要求每条道路至多被一条赛道经过。
- ▶ 目前赛道修建的方案尚未确定。你的任务是设计一种赛道修建的方案，使得修建的 m 条赛道中长度最小的赛道长度最大（即 m 条赛道中最短赛道的长度尽可能大）

NOIP2018 赛道修建

- ▶ 没想到吧，这道题也是贪心！
- ▶ 先介绍一下所有的部分分吧。
- ▶ $m = 1$ 部分分：取树上最长路径即树的直径，通过两次dfs或bfs就能求。
- ▶ $a_i = 1$ 部分分：菊花图。套上二分，对于一个答案 mid ，分两种情况讨论：
 1. 边权大于等于 mid ，自成一条赛道即可。
 2. 边权小于 mid ，贪心地两两组成和大于等于 mid 的赛道即可。这里使用`multiset`实现。
- ▶ $b_i = a_i + 1$ 部分分：链。这是经典二分问题，应该不用再啰嗦了吧。
- ▶ 分支不超过3部分分：正解的弱化版。做出这个部分分应该都能A了吧。

NOIP2018 赛道修建

- ▶ 正解是这样的：
- ▶ 钦定一个点为树根，定义 f_i 为以 i 为根的子树中，可以提供给父亲的链长。
- ▶ 接下来对某一棵以 u 为根的子树考虑，考虑赛道是怎么组成的。
- ▶ 我们对 u 分析，把子树内有关 u 的 f_v 都考虑，其实就是两种情况：
 1. 边权大于等于 mid ，自成一条赛道。
 2. 小于 mid ，先尝试两两拼成一条赛道，剩下的就作为 f_u 传给父亲继续利用。
- ▶ 其实就是贪心地不断往上合并，正解也不过就是这样而已嘛。

CF1141G Privatization of Roads in Treeland

- ▶ 给你 n 个点的无根树，每条边有颜色，一个点是好的当且仅当每条边都属于不同颜色。要你找到一个最小的 r ，使得颜色总共有 r 种，并使不好的点个数不超过 k 。
- ▶ $2 \leq n \leq 100000, 0 \leq k \leq n - 1$ 。

CF1141G Privatization of Roads in Treeland

- ▶ 显然，当 $k = 0$ 时，答案是所有点的最大度数。
- ▶ 并且，一个点度数越大，那么它变坏就越容易。
- ▶ 我们的贪心策略就隐隐出来了：根据度数从大到小排序，舍弃掉前 k 个点，让它们变坏，剩下的就让它们变好。
- ▶ 不难发现，要我们求的 r 就是第 $k + 1$ 个点的度数。
- ▶ 边的染色方案？我们通过一次dfs就可以进行染色。
- ▶ 每次尽量不与之前的相同，但是相同了也没关系，因为如果相同，那么这些点就是我们抛弃的坏点。
- ▶ 最后输出一下就完事辣
- ▶ 星期六做到的，自己不会，给大家看看。

再见

