贪心算法

最小区间覆盖

- 给定一个长度为m的区间，再给出n个区间的起点和终点，求最少使用多少个区间可以将整个区间完全覆盖。

解法

- 一般情况下按照某种顺序去考虑问题会简单的多，我们不妨将所有的区间按照左端点排序，左端点相同的按照右端点排序

- 假设当前选取了若干个区间，已经覆盖了1-S的区间，那么接下来选取的区间的左端点一定要小于等于S，于此同时，右端点越大越好。

最多不相交区间

- 有n个区间，最多选取多少的区间，他们之间两两不相交

- 每次选取开始时间最早的区间?
- 每次选取结束时间最早的区间?
- 每次选取用时最短的区间?

最少的点覆盖所有的区间

- 给你n个区间，求最少选取多少个点使得每个区间内都有点

解法

- 首先还是将区间排序，对于一个区间，选择任意一个点都是可以的，不如每次都选择最右边的点，能够尽可能的去覆盖后面的区间

字典序有关的贪心

- 给定长度为N的字符串S，要构造一个长度为N的字符串T，起初，T是一个空串，随后反复进行下列任意操作。

- 从S的头部删除一个字符，加到T的尾部

- 从S的尾部删除一个字符，加到T的尾部

- 目标是要构造字典序列尽可能小的字符串T

解法

- 我们有一个直观的想法，不断的取出开头结尾中字符比较小的一个字符放到T里面
- 但是当首尾相同的时候怎么办?
- 看下一个字符哪边小，如果还相同，就继续往里面找。直到不同为止，取小的那一边的那个字符

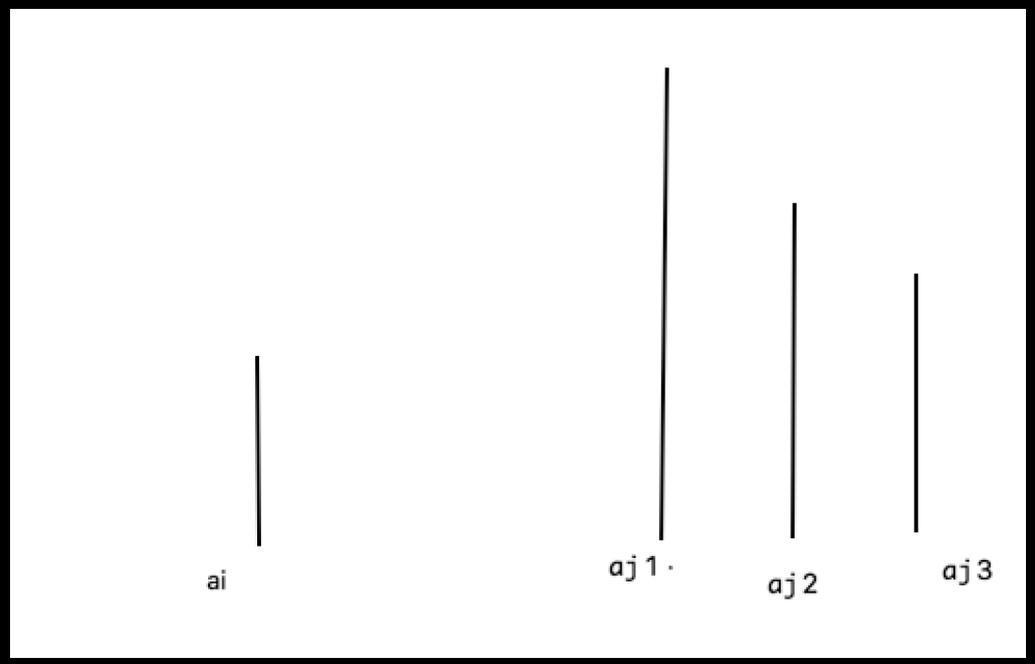
最长上升子序列

- 给你一个序列，求最长上升子序列的长度，并输出一个字典序最小的解

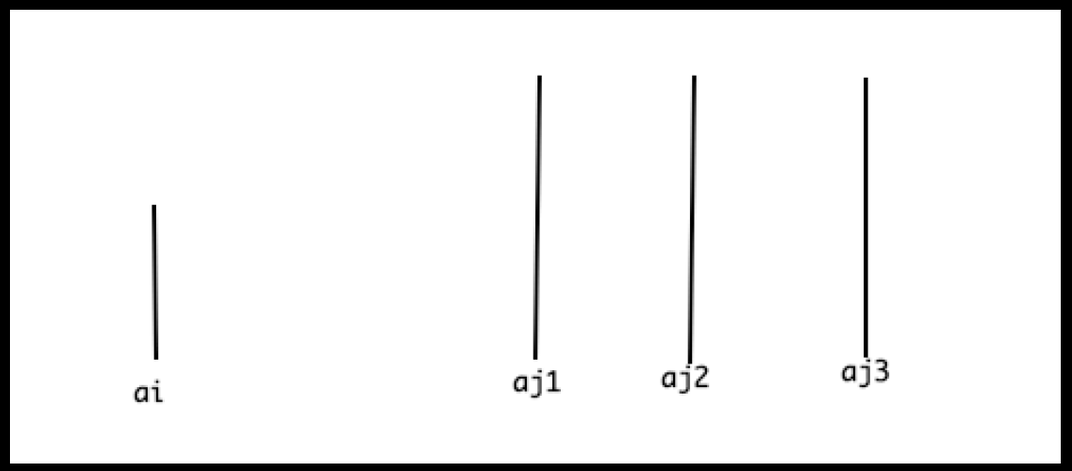
- 比如1 2 3 10的字典序大于1 2 3 9的字典序，因为前三个数相同，10比9大

解法

- 字典序问题一般是从前往后贪心，能取小的就取小的。
- 这提示我们可以从后往前dp，令dp[i]表示从i开始的LIS的长度，如果dp[i] 可以从多个dp[j](j > i && a[j] > a[i])转移过来，根据字典序最小，我们会选择a[j]最小的位置。
- 如果有多个相同的a[j]都可以转移过来，选择哪个都可以



- 有多个a值不同的dp[j]都可以转移的时候选择最小的那个显然更优



- 有多个值相同的dp[j]+1 == dp[i] 的时候，选择哪个都可以，因为后续字典序最小的解肯定在最后一个aj的后面。所有的aj的下一个数bj1 bj2 bj3…肯定是非递增的, 否则dp值就会产生矛盾。

二进制贪心

- B君和G君聊天的时候想到了如下的问题。
- 给定自然数l和r ,选取2个整数x,y满足l <= x <= y <= r ,使得x|y最大。
- 其中|表示按位或，即C、 C++、 Java中的|运算。
- 0<=l<=r<=10^18

解法

- 首先区间很大，肯定无法暴力枚举
- 由二进制的特点，我们从高到低考虑问题
- 或运算，当前位只要有一个是1就是1
- 第一个贪心：y直接取r，这样子x可选取的空间最大，如果y的有些位能取1却故意不取1，让x去取1，这样子其实是没必要的，反而可能会导致x的取值无法满足在[l，r]之间
- 第二个贪心：x从高到低考虑，如果当前位y已经为1，就尽可能取0，实在不行，取1，如果当前位y为0，就尽可能取1，实在不行取0.
-

考试安排

- 假设有n门课，每门课规定的考试时间为ai, 现在每门课都允许提前参加考试，提前的时间为 bi, 但是为了便于管理，任课老师在记录考生的考试日期的时候还是会记录ai, 孙同学每考完一门，考试系统就会增加一条记录，显示该考生在ai天的时候完成第i门课的考试(尽管孙同学可能是在bi天完成的第i门考试)，同一天可以参加多门考试，为了使得最终考试系统里面的考试日期的记录是非递减的，孙同学最快在哪一天才能完成考试。

- 假设有三门课，规定考试时间与可以提前的时间分别为
- 第一门课 6 1
- 第二门课 5 2
- 第三门课 4 3
- 显然最快也要在第6天才能完成所有考试，因为虽然可以提前考，但是记录会产生矛盾，比如第一天考第一门，第二天考第二门，第三天考第三门，记录就是6 5 4，考试日期变成了倒序

解法

- 首先观察到有两维信息，由于第一维信息一定要保持非递减，所以可以直接按照第一维关键字排序，第二维在第一维相同的情况下按照从小到大排序

- 然后我们可以扫描第二维信息，边扫描边记录考完前面的考试最少需要的天数ret，如果当前的考试提前考的日子大于等于ret，就更新ret维当前提前考的日子，否则就直接等于当前考试的规定考试时间。

小萌新与老司机

- 平面直角坐标系上有n个小萌新，与m个老司机，每个小萌新都渴望老司机的光环，一个萌新如果能被一个老司机的光环照到，当且仅当老司机的(x, y)坐标都大于等于萌新的(x,y)坐标,但是一个老司机只能照亮一个小萌新，如果某个小萌新被照到，他就会获得500*x+2*y的快乐值，问你最多有多少个小萌新能被照到以及在此基础上最多能获得多少快乐值0<x<1440, 0 < y < 100

解法

- 由于题目需要满足两维信息同时满足大小关系，直观的想法是可以将某一维按照关键字排序，那就只需要考虑接下来一维的信息了

- 排序排掉一维后，我们就可以尝试线性扫描专门统计另外一维的信息。在此题我们需要给每一个萌新找一个配对的司机。

- 观察500*x+2*y这个式子，以及x，y的数据范围我们可以得出，x增加1的效果大于y增加100的效果。因此此处可以得出我们的第一个贪心结论：在不影响整体数量的情况下，优先匹配那些x比较大的萌新。

- 我们先将萌新与司机们分开各自按照x从大到小排序，那么现在能照到某个萌新的司机肯定是司机数组中从后往前连续一段中y比他大的，这个时候可能会有多个司机满足条件，于是我们就需要去决策怎样的选择是更优秀的。
- 直观的想法是：我们将萌新和司机都看成坐标上的点，那么现在的问题转换为对于一个萌新，在他的右边高度大于等于他的区域找一个点跟他配对，我们可以用一个数组来维护右边的那些司机的高度，然后找高度大于他的最低的司机来跟他匹配，以使得那些更高的司机可以用来帮助更需要帮助的萌新。具体代码的写法可以仔细思考下。

邓哲也的矩阵

- 有一个矩阵，有一个变量sum=0，现在你需要对矩阵进行K次操作，每次操作你需要选取某一行或者某一列，将选取的行或者列的元素之和加到sum，然后对这一行或者列的所有元素都减去p，问你K次操作后，sum的最大值。

例子

- 2 2 2 2(n, m, k, p)
- 1 3
- 2 4
- Ans = 11
- 2 1 3 2
- 3
- 3
- Ans = 8

- 由第二个例子我们可以知道，直接每次贪心选取和最大的行或者列是不对的，因为元素可能会变成负数。

- 我们观察每一个格子的情况可以发现，一个格子的值要么被某一行取到，要么被某一列取到，如果选取哪些行，哪些列都确定了，那么这个格子上的值对最终答案的贡献也是确定的，他不会随着你选取行列的先后顺序而变化

- 于是我们可以得出结论，假设你选取了R行，C列，这R行，C列选取的顺序是不影响最终答案的。

- 再进一步，我们可以将行的选取全放在前面，最后一起进行列的选取。

- 最后我们可以得到解法，贪心从大到小选取每一行，然后记录f[i] 表示选取i行所能获得的最大值，列也同样处理，最后再枚举选择的行数，将行的影响算到列里面去，因为假设选了R行，那么你在之后选取每一列的时候，都要减去R*p

2015普及组T4推销员

题目描述

阿明是一名推销员，他奉命到螺丝街推销他们公司的产品。螺丝街是一条死胡同，出口与入口是同一个，街道的一侧是围墙，另一侧是住户。螺丝街一共有 N 家住户，第 i 家住户到入口的距离为 S_i 米。由于同一栋房子里可以有多家住户，所以可能有多家住户与入口的距离相等。阿明会从入口进入，依次向螺丝街的 X 家住户推销产品，然后再原路走出去。

阿明每走 1 米就会积累 1 点疲劳值，向第 i 家住户推销产品会积累 A_i 点疲劳值。阿明是工作狂，他想知道，对于不同的 X ，在不走多余的路的前提下，他最多可以积累多少点疲劳值。

输入输出格式

输入格式：

第一行有一个正整数 N ，表示螺丝街住户的数量。

接下来的一行有 N 个正整数，其中第 i 个整数 S_i 表示第 i 家住户到入口的距离。数据保证 $S_1 \leq S_2 \leq \ldots \leq S_n < 10^8$ 。

接下来的一行有 N 个正整数，其中第 i 个整数 A_i 表示向第 i 户住户推销产品会积累的疲劳值。数据保证 $A_i < 1000$ 。

输出格式：

输出 N 行，每行一个正整数，第i行整数表示当 $X = i$ 时，阿明最多积累的疲劳值。

【数据说明】

对于 20% 的数据， $1 \leq N \leq 20$ ；

对于 40% 的数据， $1 \leq N \leq 100$ ；

对于 60% 的数据， $1 \leq N \leq 1000$ ；

对于 100% 的数据， $1 \leq N \leq 100000$ 。

解法

- 这个题目核心的东西就是要观察出选i个位置的答案一定在选i+1个位置的答案里面，可以利用反证法，每次要么最右边的位置往右移了，要么在左边已走过的距离内选择一个。
- 基于上述事实，我们的做法就是记录下当前最右边推销过的住户的位置j，在左边未推销的ai里面找一个最大，在右边找一个ai+2*si-2sj最大的数，选这两个中较大的一个，所以本质上前缀后缀分别维护两个堆即可。

解法2

- 当最右边的位置确定之后，其实就是在前面选择X-1个较大的a值
- 前面的X-1个较大的a值实际上就是全局的X-1个较大的a值
- 要是有一个漏网之鱼在后面，显然选这个较大的又远的会更优
- 所以答案的形式就是两种：1，选的就是前X大的位置，2，选的是前X-1大的，加上后面2*s+a最大的一个位置。
- 所以对a的值排序就可以解决这个问题了

2013提高组Day1 T3火柴排队

涵涵有两盒火柴，每盒装有 n 根火柴，每根火柴都有一个高度。 现在将每盒中的火柴各自排成一列， 同一列火柴的高度互不相同， 两列火柴之间的距离定义为： $\sum(a_i - b_i)^2$

其中 a_i 表示第一列火柴中第 i 个火柴的高度， b_i 表示第二列火柴中第 i 个火柴的高度。

每列火柴中相邻两根火柴的位置都可以交换，请你通过交换使得两列火柴之间的距离最小。请问得到这个最小的距离，最少需要交换多少次？如果这个数字太大，请输出这个最小交换次数对 $99,999,997$ 取模的结果。

【数据范围】

对于 10% 的数据， $1 \leq n \leq 10$ ；

对于 30% 的数据， $1 \leq n \leq 100$ ；

对于 60% 的数据， $1 \leq n \leq 1,000$ ；

对于 100% 的数据， $1 \leq n \leq 100,000, 0 \leq$ 火柴高度 $\leq maxlongint$

排序不等式

已知$a_1 \le a_2 \le a_3 \cdots\cdots \le a_n$，$b_1 \le b_2 \le b_3 \cdots\cdots \le b_n$。

试证：$\sum_{i=1}^{n} a_i b_{n+1-i} \le \sum_{i=1}^{n} a_i b_{j_i} \le \sum_{i=1}^{n} a_i b_i$，其中$j_1, j_2 \cdots\cdots j_n$是 1,2 …… n的一个排列。

证法 1：

不妨设 $j_n \ne n$,(若$j_n = n$,则考虑j_{n-1})，第m项 $a_m b_{j_m}$中 $j_m = n$

那么在乱序和S中，调整第n项和第m项位置，易知 $a_n b_n + a_m b_{j_n} \ge a_n b_{j_n} + a_m b_{j_m}$，故新的和 $S_1 \ge S$。

同理调整第 n-1 项，得到$S_2 \ge S_1 \ge S$，最多经过 n-1 次调整可得：

$a_1 b_1 + a_2 b_2 + \cdots\cdots + a_n b_n \ge a_1 b_{j_1} + a_2 b_{j_2} + \cdots\cdots + a_n b_{j_n}$

显然，当且仅当$a_1 = a_2 = a_3 \cdots\cdots = a_n$，$b_1 = b_2 = b_3 \cdots\cdots = b_n$时取等。

不等式右边得证，同理可证左边也成立。

```
a b
c d

ad + bc  vs  ac + bd
a (d - c) - b (d - c)
(a - b) * (d - c) < 0
```

解法

- 可以固定一个序列不动，将另一个序列调整到与这个序列每个位置的数的排名都一样

- 重新规定顺序后，本质上就是在算逆序对

总结

- 贪心算法最关键的地方，要说服自己这样做为什么是最优的，为什么不存在更优解了。

- 常用的一些方法有，分类讨论所有情况，反证法等。举出大量的例子，打表等手段都可以用来辅助证明贪心的正确性。在比赛的时候需要大胆猜测，小心验证。