贪 心 导 论

By Envelope2

目 录

1. 什么是贪心

在算法与思想不断多样化的 OI 世界里，有一样东西。它既是一种思想，一种普及大众、简单快捷的思想；又是一种算法，一种几乎无所不能、高效率、经济实用的算法。它是 OI 新手们忠实崇拜的偶像，又是 OI 大牛们理想的最高境界。没错，这个东西，就是————贪心。

常言道：“贪得无厌。”然而，面临多元化的 OI，我们不得不选择贪心。为什么？因为贪心是紧随时代潮流的。曾经有一位 OI 贪心大牛说过，“给我一道题，我就能贪心”，这就是最高境界。

有人会问：“贪心究竟是什么？”我无法准确回答，因为没有标准答案。在每个人心目中，贪心的定义是不同的。思想不同，算法也不同；但有一个共同目的————**将思想简单化**。

曾经有位大牛写过一篇《骗分导论》，我很喜欢。在我看来，贪心就是一种高水平的骗分技术。“有技术，一等奖也贪得来”，这就是我的信仰。

还有人说：“贪心很困难。”正因如此，今天，我写下这篇《贪心导论》，以此与各位 OIer 们共享经验教训，也当作是考前的娱乐消遣。

2. 贪心的实际应用

今天，OI 崇尚的是 **动态规划、搜索** 相结合。然而，我却要在这里说贪心，原因何在？在于 **贪心的广泛性** 。

首先不得不说明的是，本人的贪心技术还不够到位，因此，本人对贪心的定义就是： **尽可能地多拿分** ，而非以 AC 为目标。

举几个简单例子来说明一下。

[例 1]有 N 个人排队接水。每个人接水所用的时间不同，而一个时刻只能有一个人接水，且只有等一个人接完水后，另一个人才能开始接水。现在请你求出 N 个人接水的顺序，使得每个人所用平均时间最短，并输出这个最短平均时间。

这道题是贪心的经典例题了。当初此题发布时，有人想当然地认为是动态规划去求解。虽然此题用动规也能解决，但是有没有更简单的方法呢？——当然有，就是贪心。联系一下生活实际：排队的时候，如果出现拥挤情况，怎样做才能节约时间呢？让耗费时间最短的人先来，还是让耗费时间最长的人排第一？这样一想，问题就变得简单了，贪心策略也出来了————**让耗费时间短的人排在前面**。于是一个快排，轻松解决。看看这效率，那叫一个高啊~~~~~~~~~~

[例 2]经典背包问题。一个体积为 V 的包，现在有 N 个物品，每个物品都有一个体积和一个价值。现在让你求出一个最佳方法，使得放进背包里的物品的价值尽可能大（当然放进去的物品总体积不会大于背包体积）。

这个问题我想就不用我废话了吧？也是经典的贪心问题，一种较优的策略为：**将性价比高的物品尽可能多地往里放**。虽然由于数据不会太弱而难以 AC，但分肯定是多拿定了。而且，通过贪心，有效节约了时间，用以分配到其他难题上，可谓两全其美。

从以上例子中可以看出贪心在 OI 实际中的广泛应用了。总之一句话：

贪心才是王道！掌握贪心，与掌握动规同样重要！

3. 常用贪心方法（策略）

接下来就是本人的自主品牌了~~~~~经过长时间的摸爬滚打，本人研究出一套常用贪心方法（策略），个人感觉很实用。本人曾凭借它成功拿下过 773 分。

值得一提的是，贪心在大多数情况下只是一种思想，因此需要一种算法作为媒介才能得以实现。而我所做的，不过是帮其匹配而已。

No.1 裸搜 + 贪心 = 裸贪

不要误会，名字纯属娱乐。说得直接点，就是贪心的思想，用深搜（不加剪枝的深搜称作“裸搜”）来实现。不要被吓到了。这是最基础也最常用的贪心策略，关键时刻用它足以左右命运。

[例 1]著名的“炒股票”问题。现在你手里有 1000 元，在接下来的 N 天中，你可以选择将手中的现金按当天价格购买股票，也可以将手中的股票兑换成现金。已知第 i 天的股票价格为 a[i]（单位：元/股）。现求出第 N 天时手里能拥有的最多现金数量。1<=N<=15。

这道题很简单，就是一个简单动规。不过如果有的人用动规做不出来的话（只是“如果”而已，我并不是贬低各位水平~~~），就得考虑——贪心。

首先很容易想到：每天 **要么全部买进或卖出，要么什么都不做** 。因为每天要么会赚（要赚就要多赚点，于是全买或卖），要么不会赔（什么都不做就不会赔了）。有了这一点，其实动规已经很容易了。不过呢~~~我就是要贪！怎么办？注意题目的数据范围。考虑到 N 的值很小，在深搜可以承受的范围内，于是我们可以用 裸贪来解决这道题了。策略就是：**搜索每一天的两种决策，不断替换最优**。一样 AC。

[例 2]现在有 N 件工作。每件工作可能会依赖于其他工作（该工作必须等其所依赖的工作全部结束后才能开始，一个工作最多依赖于 10 件工作）。假设 1 个时刻只能完成一个工作，现在求出第 M 件工作最早完成的时间。数据保证没有循环依赖。

初看题，是一个多叉树结构。有的同学一看跟树有关就甩掉不做了（俗称“树型恐惧症”）。其实仔细一考虑：没有环，数据也不是太大 ••• 那么，**从目标点出发搜索所有依赖对象直到目标点的依赖对象全部完成为止**，就是一个简单的裸贪问题了，根本用不着什么树。（当然你要用树我也没办法~~~）

由上能够看出裸贪的高效率。其实，对于一些复杂的动态规划和广搜题目，用裸贪都能取得一定的成效（虽然不一定能 AC，比如上面例子中 N=1000~~~~）。对于考试中的应急措施， 裸贪 可谓首选。

No.2 枚举 + 贪心 = 枚贪

仿照上面来理解的话就是用枚举的方法来实现贪心。这个实现起来就更容易了。枚举嘛，N 重循环都能搞的定的东西。可惜的是，效率不太高，想要 AC 不容易。

[例 1]小 S 一天一餐只吃 N 顿饭。每顿饭会有 M 份食物供其选择。这些食物又被分成 K 类。根据

小 M 的爸爸规定，小 M 每顿饭吃第 i 类食物的数量不能超过 a[i]。每份食物都有一个营养值。问小 M 应该怎样安排饮食，使得每顿饭可以获得更多营养值。

有的 OIer 们一看此题，条件反射般地联想到背包问题，于是不亦乐乎地动态规划 ing~~~~停！既然背包问题可以贪心，这个题为什么就不能呢？其实就是一种最低级的贪心策略：**每次找能够吃的食物中营养值最大的一份**。还是一个快排，挨个枚举，搞定！很简单吧？不过有的大牛们反而会被这种简单题给迷惑，把思想复杂化了。所以，简单就好。

[例 2]现在有 N 个数，要求前 M 个数的乘积刚好等于后 N-M 个数的乘积，求这样的数序列有多少个。1<=N<=8。

这是一道典型的优化深搜题。但是在考场上，有的同学们只能想到深搜，而想不到优化（即 裸搜 ）。而裸搜是肯定要超时（或者爆掉）的。怎么办？用 枚贪或许可以帮助解决此题。

仔细思考一下：既然搜 8 个数会挂，那么搜 4 个行不行呢？如果行，剩下 4 个又怎么办呢？因此，我们想到：**只搜索前 M 个数，枚举后 N-M 个数；且当后 N-M 个数之积已经大于前 N 个数之积时果断退出。**这样一来，时间复杂度虽然没有太大优化，但空间复杂度大大降低了（搜索深度降低了）。当时的我因为 RP 爆发，用 枚贪 一次 AC，实乃幸运（其中有一组是卡着 1s 的时间过的~~~~）。

在实际操作中，枚贪 容易实现但多半不能 AC，过 5、6 组就是运气了。

No.3　模拟　+　贪心　=　模贪

用 模拟 来贪心，是一种稍高级的方法。

[例]著名的俄罗斯方块游戏大家都玩过吧？现在有一种简单的方块游戏：有宽度为 W 的屏幕，有 N 个边长不一的正方形方块先后从上往下掉，且每一步在保证最大高度尽可能小的情况下尽可能地向左放。现给出每个方块的边长，请你求出方块掉落完后的最大高度（假设屏幕高度无限）的最小值。

又是一道蛊惑人心的题。题目最后的“最小值”一词是一个陷阱，让人瞬间误认为是求最优解问题，从而考虑动规。其实按照题目描述理解的话，此题的答案相对于数据来说是惟一的，贪心策略也显而易见了：**每步累加每个宽度对应高度的最大高度，从中找最小的掉落下来**。简单的模拟就能搞定。

因为用 模拟 的方法来贪心的时候并不多，因此就不多讲了。大家有兴趣可以亲身体会一下。

No.4　动规　+　贪心　=　动贪

本年度最高级也最深奥的贪心方法！动规本身就难，还要用动规来贪心，更是难上加难！我向来最讨厌动规，但对于 动贪 还是不得不罗嗦几句。没办法，老大啊！

[例]现在你安排 N 个人吹 M 个气球。已知第 i 个人每分钟可以吹 a[i]个气球，吹一分钟后要休息 b[i]分钟。现在求吹完 M 个气球所用的最短时间。当然，每分钟只能有一个人吹气球。(1<=b[i]<=4)

当初做这道题时，想到低级的 枚贪（每次找能吹的人中吹的最多的一个） ，但是只过了 9 组。万般无奈之下参看标程（当然是纯动规了，6 维的），发现标程果然很经典!因为休息时间最多也才 4 分钟，

因此我们想到一个超高级的贪心策略：**枚举后 4 分钟吹气球的人，借以递推求解**。这样一来就完全没有了反例，也不会超时，虽然麻烦了点，但是绝对的 AC！

动贪 的效率是其他贪心方法所远不能及的，也是几乎可以保证 AC 的贪心方法。 编程复杂度 是它的惟一缺陷，因为很有可能在考场上你多半想不出要 动贪 ，即使想出了，冗长的代码也会令你望而却步。正因如此， 动贪 才成为了大牛们的终极目标之一。对于菜菜的本人来说，就算了吧`~~~~掌握好以上三种贪心就行了。

No.5 树论 + 图论 + 贪心 = 树图双贪

这是一种较少见的贪心方法，因为遇到图论首先想到的是 Krim、Floyed、SPFA 等高级算法，而不会首选贪心；只有在无可奈何的情况下，才会选择 树图双贪 来凑合凑合。虽然是下策，但是不到最后，还不会知道会发生什么呢~~~~~~~~~~~~~~~~

[例]二叉树有很多种，但是它们很多都是等价的。某二叉树通过把某非根结点作为根结点而保持其它各结点之间的关系不变，重新建立出一棵树，若这棵树是一棵二叉树，则称这两棵二叉树等价。现在给出一棵基准二叉树，请你判断给出的其他树是否和它等价。（图略，因为对于贪心来说不重要~~~）

题目叙述相当复杂，即使有示意图，部分 OIer 们也是一头雾水，于是毅然放弃。不过没关系，即使不懂题意，仍然可以贪心，因为 贪心无极限 嘛！此题应该如何贪心呢？

首先想到一个树的常用术语——度。既然只是判断是否等价，那么，**统计所有结点的度并比较**可不可以呢？肯定是有一定根据的。这是最简单的贪心策略，效率达到了 通过 9 组 之高！这是树型贪心。

再考虑图的常用算法：本题的二叉树变形只是基于各结点间的相互位置，因此我们想到：**利用 Floyed 算法求出两点间的最短距离再加以判断**。这就是 图论贪心 了。虽然不太好理解，但是它的效率说明了一切：AC！我们不得不服气了。贪心就是如此，搞得你很郁闷却又不得不接受现实。

关于 树图双贪 ，就点到为止。如果哪位大牛有意愿与本人交流一下此方面的问题（因为我是不擅长树论和图论的，所以自然不会用此方法~~~~），请发言。本人倒是兴致勃勃呢~~~

No.6 乱搞 + 贪心 = 乱贪 ？

最低级的骗分伎俩，纯属贪图速度，因此效率极低~~~~~当然，乱搞无语，贪心有理，我们还是要见识一下 乱贪 在关键时刻的威力。

[例]（某次模拟赛的一道题）现在老师交给小 X 一个光荣而艰巨的任务：在规定的 T 时刻前完成若干项任务。给出每项任务的最迟完成时间和完成所需时间，请你判断小 X 能否顺利完成任务。因为老师很 BT，而且小 X 的 RP 极低，因此任务的数量可能会超过 Maxlongint。一个时刻只能做一件任务，并且要做就要做到底。若能完成，输出“Winner”；否则，输出“Beaten”。

此题的数据相当之 BT（高精度是肯定的），繁杂的数据处理几乎令每一个 OIer 崩溃。此时，无论用前面所讲的什么贪心方法，似乎对此题的复杂度都起不到太大作用。那么，现在的我们，是否就一筹莫

展了呢？————————————还没有结束。事已至此，就别怪我更 BT 了————乱贪！

由于乱贪的方法太多（想得到的都算），这里只介绍两种比较 BT 的：

1．**If 任务数<=T then writeln('Winner') else writeln('Beaten') .**可以得 20 分；

2．**Writeln('Beaten')**。我当时是这么想的：此次模拟赛应该有一定的恶搞成分，题目中的小 X 应该是出题人的某位好友；既然如此，借此机会，何不捉弄他一番？反正只是娱乐而已。于是，数据中结果为 Beaten 的~~~~~~~~~~~~~~~~~~~~~竟有 8 组！于是 80 分到手了。（什么题啊这是~~~~~）

出现这种状况，只能怪 RPWT 了。毕竟人心变幻莫测，在结果之前你永远不知道接下来会发生什么。因此，　乱贪　仅限于保命用，平时尽量不用。（详情请参考《RP 导论》以提高 RP）

4. 贪心技巧

说了这么多，本人要强调的只是 **贪心思想的有效运用** ；也许你会认为我上面所讲的并不是以 贪心 为重点————————本来就不是。贪心只是桥梁而已，真正发挥关键作用的还是日常积累的算法。所以，不要被贪心所引诱从而放松日常算法的学习加深，否则后果很严重（猜猜看，谁是受害者？）。

有的人认为贪心很难，难在思想。算法是无穷的，题目是多样的，要在数不清的题目面前，找出数不清的贪心策略，从中选择最优，并从数不清的算法中，搭配最合适的一种 •••••• 想想都很恐怖了。为此，在 NOIP2007 的前夕，本人仅持个人观点，向各位小牛（大牛就不用听了~~~）们提出以下建议：

1. 遇题先不要忙着贪心。一些数学问题是贪心所不能及的；一些送分题又无须贪心就能解决的。先中规中矩地处理掉简单题，再考虑从 贪心 下手。

2. 先枚举，后模拟，实在不行用搜滴~~~~也就是说，实在不行要贪心的话， 枚贪 是首选（方便哪！）；如果数据的规模超出了枚举的极限，就考虑 模贪 （稳扎稳打，不费劲）；如果数据的复杂度相当高，以至于超出模拟的承受规模，就只有忍气吞声，用 裸贪 （最好加点优化。反正我没有那个习惯~~）以求得分了。通常情况下，一次成功的 枚贪 可以拿到 50 或 60 分，一次成功的 模贪 也差不多 70 分左右；当数据规模较小时， 裸贪 有 AC 的可能，一旦规模 BT 起来~~~~~30 分就足够了。————————至于 动贪 等高等贪心，有本事的人当然就用，没本事的就老实点吧~~~~最后重申，不推荐 乱贪 ！

3. 越是贪心，越要细心。虽然贪心能够降低编程复杂度，但是我们决不能掉以轻心。最简单的程序都实现不了，高难度的又怎么办？本人最初学习贪心时，每一次几乎都能找到合适的策略，但总是一败涂地。原因就在于 **全局变量与局部变量混淆、函数与过程的混淆、变量太多引起混淆** 等一系列基础错误。平时，可能会有一天的时间来调试一道题目；上了考场，连 贪心 都要调试两个小时，那你还能做什么？贪心也讲究扎实的基本功。

4. 注重思维锻炼，提高思维的活跃性。贪心是一门艺术，也是对大脑思维的考验。通常情况下，思维越活跃，视野越开阔，联想越丰富，从而对贪心起到了极大的促进作用。所以有调查显示，越是捣蛋分子，越能贪！（举个思维活跃的例子：**我的钱包哪儿去了？找一下：先枚举几乎所有可能的地方，如果是白费劲，然后静下心来模拟一下“案发现场”；如果记忆力衰竭的话就来个大搜索，一个死角也不放过！**）

5. 加强记忆化的能力。众所周知，无论是动态规划还是搜索，记忆化思想是很必要的。贪心同样需要记忆化，因为一种贪心方法如果能够对付一道题，那么对于相类似的题，也应该行的通。记忆化，已经成为大众化的需求。

NOIP2006 已经过去，NOIP2007 即将来临。做好知识储备，带上算法积累，保持心态良好；最后，以贪心为本钱，勇往直前！Believe it or not, 我会做的更好！

END.