浅析一类二分方法

（IOI2012 中国国家集训队第二次作业自选部分）

哈尔滨市第三中学 王钦石

摘要

二分是算法竞赛乃至所有算法设计中常用的方法，本文介绍了一种方法，把一类看似与二分毫不相关的限制个数的最优化问题通过二分转化为不限制个数、可以用简单高效的算法解决的问题，从而高效的解决限制个数的问题。

关键词：二分，单调性，限制个数

几个简单的问题[1]

【例 1】

给定一个正整数列 $a_0..a_{n-1}$，分成若干段，记每段之和为 S_i，求最小的 ΣS_i^2。

显然，把每个数分为一段答案最优。

【例 2】

给定一个正整数列 $a_0..a_{n-1}$，分成若干段，记每段之和为 S_i，求最小的 $\Sigma(AS_i^2+BS_i+C)$。（A,B,C 均为正整数）

很容易得到一种动态规划算法，f(i)表示前 i 个数的最优答案，枚举最后一段的长度进行转移，时间复杂度为 O(N^2)。

记前 i 个数之和为 sum_i，$g(i) = f(i)+Asum_i^2-Bsum_j$，转移时只需找出使 $g(j)-2Asum_isum_j$ 最小的 j 转移即可，这可以用凸壳进行维护，在 O(N)的时间内计算出答案。

【例 3】

给定一个正整数列 $a_0..a_{n-1}$，分成若干段，记每段之和为 S_i，求最小的 $\Sigma(AS_i^3+BS_i^2+CS_i+D)$。（A,B,C,D 均为正整数）

朴素的动态规划算法依然成立，不过不能按例 2 的做法进行优化。记 p(i)为 f(i)决策中最优的 j 中最小的一个，显然对于 i>j，有 p(i)≥p(j)。换而言之，对于 i>j，存在一个 x，使得对于所有 k≥x，f(k)决策中 i 更优，反之 j 更优。所以可以维护一个双端队列，依次计算每个 f(i)。核心部分用一下伪代码描述：

```
Deque q
For i = 0->n-1
    While 对于当前 i，q 中的第二个元素优于第一个
        q.pop_front()
    计算 f(i)值
    While i 比 q 中最后一个元素优时，q 中最后一个元素尚不优于倒数第二个元素(内部需使用二分，复杂度 O(logN))
        q.pop_back()
    q.push_back(i)
时间复杂度为 O(NlogN)
```

[1]这部分内容较为简略，如有不理解的地方可参照周源的论文。

限制个数的问题

【例 4】

给定一个正整数列 $a_0..a_{n-1}$，分成不超过 K 段，记每段之和为 S_i，求最小的 ΣS_i^2。

显然，分为少于 K 段是不优的。

仍然可以使用动态规划法，f(i,j)表示将前 i 的数分为 j 段的最优答案，枚举最后一段的起始点进行转移，时间复杂度为 $O(N^2K)$。

同样可以利用决策单调性进行优化。记 p(i,j)为使 f(i,j)最优的决策中最小的一个，易见 $p(i-1,j) \le p(i,j) \le p(i,j-1)$，可以控制顺序使得 f(i-1,j)和 f(i,j-1)先于 f(i,j)计算，这样对于每个 x，所有 i+j=x 的状态总共只需考察 O(N)个转移，总时间复杂度降为 O(N^2)。

另一种优化方式是按 j 从小到大计算，i 从小到大依次计算，用例 2 中的方法维护一个凸壳，在均摊 O(1)的时间内计算出每个状态，总时间复杂度为 O(NK)，如果认为 K 与 N 同数量级，与上一个算法无异，如果 K 较小，这一算法效果很好。

这类动态规划算法需要至少 O(NK)个状态，所以 O(NK)已经是这种算法的下界了。要想得到更好的算法，只能另辟蹊径。贪心是一个好的想法，但是很遗憾我没有找到一个保证正确的贪心或类似的算法，不过在随机数据中，贪心表现很好。

注意到动态规划算法中有一些 f(i,j)是明显不优的，可以利用一些方法去掉一些状态来降低复杂度。这种做法缺少解的质量或时间复杂度的保障，而且是非常不优美的，我没有做详细的研究，不过对于一道 OI 题目来说，这种方法还是可以拿到比较好的分数的。

下面来讲我的做法：考虑例 2，令 B=0，A=1，对于某个 C，如果最优的划分方案恰好分为 K 段，那么此答案减去 C*K 即为原问题的答案。那么我们只需求出一个恰好分为 K 段的 C 即可。显然，C 值越大，最优解中划分的段数越小，所以我们只需二分 C，获得一个恰好分为 K 段的 C 即可。时间复杂度为 O(nlogC)，其中 C 为某一常数。

问题看似已经得到了圆满的解决，不过这里还有这样一个问题，可能不存在恰好分为 K 段的 C 值。这可以通过一点修补解决，如果发现 C 取 x 时，所有最优解中最小的分段数大于 K，而取 x+1 时最小的分段数小于 K，那么 C 取 x+1 时也存在分段数为 K 的最优解，只需取此时的最优答案减(x+1)*K。

另一个限制个数的问题

以上方法更早被用于这一问题：

【例 5】

给定一张带权图，求某点度数恰好为 K 的最小生成树。

这一问题可以加强为如下问题：

【例 6】

给定一张带权图，每条边被染为黑色或白色，求白边数目恰好为 K 的最小生成树。

（题目来源：2012 年国家集训队命题互测 陈立杰[2]）

给每条白边的权值加 x，并在黑白边权值相等时优先取黑边，求出最小生成树。二分得到使最小生成树中白边数不超过 K 的最大 x，x 取此值时的只考虑权值的最小生成树中一定存在一个白边数恰为 K 的（在白边最少的生成树中不断把黑边替换为白边），时间复杂度 O(MlogMlogC)。

可以做一些优化。做出只用白边和只用黑边的最小生成树，这样可以把边的数目降至 O(N)。先将黑白边分别排好序，每次归并出整体的顺序，时间复杂度 O(M+N(logN+logC))。

存在问题的证明

例 4 中存在一个问题：固定数列 a，记 ans(i)为将该序列分 i 段的最小平方和；若存在 i<j<k 使得$\frac{\mathrm{ans(i)-ans(j)}}{\mathrm{j-i}} < \frac{\mathrm{ans(j)-ans(k)}}{\mathrm{k-j}}$，算法将无法正确计算 ans(j)。所以我们需要证明不存在这样的情况，这等价于 ans(i-1)-ans(i) ≥ ans(i)-ans(i+1)。这一点看起来应该是对的，我先是写了一个程序检验这一点，随机了大量的数据没有发现反例（对于解题来说这就够了）。证明也很简洁，不过很难想。

考虑数列 $a_0, a_1 \ldots a_{n-1}$, x，其中 x 为正实数。记对于将此数列分为 i 段的答案为 ans’(i, x)，最优方案中最后一段的起始点最小为 p’(i, x)(即最小决策点)。注意到 ans’(i, x)关于 x 连续，且仅在最优方案中最后一段大小不唯一处不可导，这样的 x 值只有有限个。$\frac{\mathrm{d\,ans}(\mathrm{i},x)}{\mathrm{dx}} = \mathrm{x} + \mathrm{sum_n} - \mathrm{sum_{p'(i,x)}}$，注意到 p’(i, x) ≤ p’(i+1,

[2] http://oj.tsinsen.com/ViewGProblem.html?gpid=A1313

x)，所以$\frac{d\,ans(i,x)}{dx} \geq \frac{d\,ans(i+1,x)}{dx}$，所以 ans'(i, x)-ans'(i+1, x)随 x 增大不降。注意到

$$\lim_{x\to 0}\left(ans'(i,x) - ans'(i+1,x)\right) = ans(i) - ans(i+1)$$

$$\lim_{x\to \infty}(ans'(i,x) - ans'(i+1,x)) = ans(i-1) - ans(i)$$

所以

$$ans(i-1) - ans(i) \leq ans'(i,x) - ans'(i+1,x) \leq ans(i) - ans(i+1)$$

得证。

一些相关问题

给定一个正整数列$a_0..a_{n-1}$，分成不超过K段，记每段之和为S_i，求最小的$\Sigma f(S_i)$。其中 f(x)满足四边形不等式 f(a)+f(b) ≤ f(a-c)+f(b+c), a≤b, c ≥ 0。

对于此问题，例 4 的算法依然成立。

猜想可以在很好的时间内解决如下问题：

给定数列 $x_{0..n-1}$ 和 $a_{0..n-1}$，其中 x 单调不减，在[0, n)的整数中取 K 个形成集合 S，满足 0∈S，求$\sum_{i=0}^{n-1}\left(x_i - \max\left\{x_j \middle| x_j \leq x_i, j \in S\right\}\right)a_i$的最小值。[3]

给定数列 $x_{0..n-1}$ 和 $a_{0..n-1}$，其中 x 单调不减，在[0, n)的整数中取 K 个形成集合 S，求$\sum_{i=0}^{n-1}\min\left\{\left|x_i - x_j\right| \,\middle|\, j \in S\right\}a_i$的最小值。[4]

[3] 此题中限制车站个数的情况 http://oj.tsinsen.com/ViewGProblem.html?gpid=A1251

[4] 试题来源：IOI2000http://poj.org/problem?id=1160

参考文献

《算法艺术与信息学竞赛》刘汝佳、黄亮
《动态规划算法的优化技巧》毛子青（IOI2001 国家集训队论文）
《浅谈数形结合思想在信息学竞赛中的应用》周源（IOI2004 国家集训队论文）
《Tree 解题报告》陈立杰（IOI2012 国家集训队第二次作业）