

# 第七届（2001）分区联赛复赛解题报告

## (提高组)

俞玮 赵爽

### 第一题：一元三次方程求解

给出一个三次方程  $f(x) = ax^3 + bx^2 + cx + d = 0$ ，试求它所有的三个根。这里所有的根都在区间  $[-100, 100]$  中，并保证方程具有三个实根，且它们之间的差不小于 1。

#### 分析：

如果是一般的求三次方程根的问题，那么只能直接使用求根公式，但这是非常复杂的。由于题目要求只精确到 0.01，故我们考虑一下是否可以应用数值方法进行计算。由题目给出的方程在区间内存在根的条件可知，我们可以用一个变量  $i$  从 -100.000 到 100.000 以步长 0.001 做循环。若  $f(i) \cdot f(i + 0.001) < 0$ ，则可知在区间  $(i, i + 0.001)$  内存在方程的解。

这样无论这个区间内的解是什么，在取两位小数的精度后都与  $i$  取两位小数的结果是一样的。故我们就可以把取两位小数精度的  $i$  作为问题的解。另外还有可能方程的根在区间端点  $i$  的情况，我们可以通过判断  $f(i)$  是否为 0 来获得。

但这种方法由于用到了题目所要求的数值精度小的特殊性，故无法扩展。而在用数值方法求方程根的时候，我们最常用的方法就是二分法。该方法的应用在于如果确定了方程  $f(x) = 0$  在区间  $(a, b)$  内如果存在且只存在一个根，那么我们可以用逐次缩小根可能存在

范围的方法确定出根的某精度的数值。该方法主要利用的就是题目所给的若在区间  $(a, b)$

内存在一个方程的根，则  $f(a) \cdot f(b) < 0$  这个事实，并重复执行如下的过程：

- 取当前可能存在解的区间  $(a, b)$  ；
- 若  $a + 0.001 > b$  或  $f\left(\frac{a+b}{2}\right) = 0$  ，则可确定根为  $\frac{a+b}{2}$  并退出过程 ；
- 若  $f(a) \cdot f\left(\frac{a+b}{2}\right) < 0$  ，则由题目给出的定理可知根在区间  $\left(a, \frac{a+b}{2}\right)$  中，故对区间  $\left(a, \frac{a+b}{2}\right)$  重复该过程 ；
- 若  $f(a) \cdot f\left(\frac{a+b}{2}\right) > 0$  ，则必然有  $f\left(\frac{a+b}{2}\right) \cdot f(b) < 0$  ，也就是说根在  $\left(\frac{a+b}{2}, b\right)$  中，应该对此区间重复该过程。

最后，就可以得到精确到 0.001 的根。

再考虑什么样的区间内会有根<sup>1</sup>。由于题目给出了所有的根都在 -100 到 100 之间，且根与根之间差不小于 1 的限制条件，故我们可知，在  $[-100, -99)$ 、 $[-99, -98)$ 、……、 $[99, 100)$ 、 $[100, 100]$  这 201 个区间内，每个区间内至多只能存在一个根。这样对除去区间  $[100, 100]$  外<sup>2</sup>的其他的区间  $[a, a+1)$ ，要么  $f(a) = 0$ ，要么  $f(a) \cdot f(a+1) < 0$  时这个方程在此区间内才有解。若  $f(a) = 0$ ，显然  $a$  为解；若  $f(a) \cdot f(a+1) < 0$ ，则可以利用上面所说的方法求出解来。这样就可求出这个方程的所有解。

最后是输出的解要求排序。我们既可以在求出来之后排序，也可以从小到大的顺序求解，就可以按照升序求出所有解。

**数据：**

	输入	输出
1	1 -2 -1 2	-1.00 1.00 2.00
2	1 -4.65 2.25 1.4	-0.35 1.00 4.00
3	1 10 -1 -10	-10.00 -1.00 1.00
4	1 -1.8 -8.59 -0.84	-2.10 -0.10 4.00

## 第二题：数的划分

求把一个整数  $n$  无序划分成  $k$  份互不相同的正整数之和的方法总数。

<sup>1</sup> 更确切地说是只有一个根。

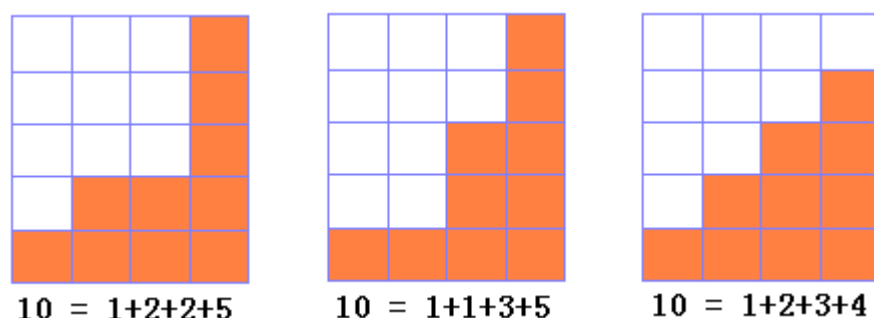
<sup>2</sup> 该区间显然只有一个数，可以用判断  $f(100)$  是否为 0 的方法来求出该区间内方程的根。对此，我们特殊处理。

## 分析：

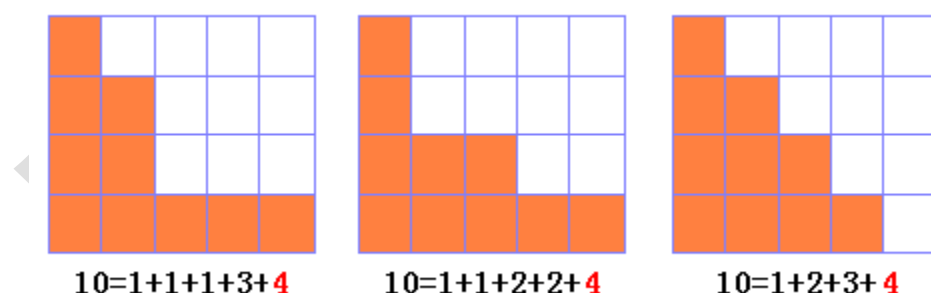
这是一道整数剖分的问题。这类问题的数学性很强，方法也很多。这里介绍一种较为巧妙的办法。

我们不必拘泥于原问题如何求解，而把思维转换一个角度来考虑一个与原问题等价的问题。

我们可以形象的把  $n$  的  $k$ -自然数剖分看作把  $n$  块积木堆成  $k$  列，且每列的积木个数依次递增，也就是这  $n$  块积木被从左到右积木被堆成了“楼梯状”。比如，下图是 10 的几种 4-剖分。



现在，我们不妨把这三个图顺时针旋转 90 度，成为<sup>3</sup>：



不难发现，旋转之后的模型还是 10 的剖分，不过约束条件有所不同。很明显，由于原来是  $k$  剖分，因此新的模型中最大的一个元素必然是  $k$ 。而其余的元素大小不限，但都不能大于  $k$ 。因此问题转化成了求  $n$  的任意无序剖分，其中最大的元素是  $k$ <sup>4</sup>。当然，我们可以把  $n$  减去  $k$ ，成为  $n'$ ，剩下的问题就是求  $n'$  的任意剖分，且其中每个元素都不大于  $k$  的方案总数了。

求解这个新的模型可以用递推的方法。用  $f(a, b)$  表示把  $b$  做任意剖分，其中最大的一个部分恰好是  $a$  的方案总数。用  $g(a, b)$  表示把  $b$  做任意剖分，其中最大的一个部分不大于  $a$  的方案总数。那么，有：

$$\begin{cases} f(a, b) = g(a, b-a) \\ g(a, b) = \sum_{i=1}^a f(i, b) \end{cases}$$

<sup>3</sup> 该图为数学上的一个著名图示（Ferrer 图），对此有兴趣的读者可以自己参看一些数学书。

<sup>4</sup> 由于篇幅有限，这里就不严格证明这两个问题的等价性了。

消去  $f$ ，有：
$$g(a,b) = \sum_{i=1}^a f(i,b) = \sum_{i=1}^a g(i,b-i) = g(a-1,b) + g(a,b-a)$$
。我们可以按照  $a$ 、 $b$  递增的顺序计算  $g(a,b)$  的值，这样在在计算  $g(a,b)$  的时候， $g(a-1,b)$  和  $g(a,b-a)$  都已经得到，故我们只用进行一次简单的加法运算即可。最后的  $g(n,k') = g(k,n-k)$  即为所求。

这种方法的时间复杂度为  $O\left[\frac{(2n'-k+1)k}{2}\right] = O\left[\frac{(2n-3k-1)k}{2}\right]$ 。空间复杂度为  $O(nk)$ ，或者我们可以用滚动数组的方法降到  $O(n)$ 。  
 该题中模型转化的思想，是很值得借鉴的。

数据：

	输入	输出
1	7 2	3
2	20 4	64
3	100 5	38225
4	200 5	583464
5	200 6	4132096

### 第三题：统计单词个数

给出一个长度不超过 200 的由小写英文字母组成的字符串（约定：该字符串以每行 20 个字母的方式输入，且保证每行一定 20 个）。要求将此字符串分成  $k$  份 ( $1 < k \leq 40$ )，且每份中包含的单词个数加起来最大（每份中包含的单词可以重叠。当选用一个单词之后，其第一个字母不能再用。例如字符串 this 中可以包含 this 和 is，但是选用 this 之后就不能再选 th）。

分析：

这一题应该算是一道比较原创的题目。注意到题目中有一个非常特殊的地方，就是以串中某个位置的字母为首字母，最多只能分出一个单词。由于在拆分字符串的过程中，如果以某位置为首某个较短单词被截断，那么以该位置为首的较长单词必然也会被截断。也就是说，对于各个位置来说我们选取较短的单词总不会比选取较长的单词所形成的单词少

这样我们可以定义一个在位置  $i$  的参数  $mlen[i]$  表示以位置  $i$  的字母为首字母，所能形成的

最短单词的长度。这样如果在这个位置加上这个单词的长度之内截断，则以该位置为首字母就不能形成单词，否则就可能形成一个单词<sup>5</sup>。这样对于所有的不同  $l$ <sup>6</sup> 个首位置，我们只要在各个位置依次对各个单词进行匹配就以得出所对应的  $mlen$  的值，这一部分的复杂度为  $O(wl^2)$ <sup>7</sup>。然后是计算把字符串分为多个部分的过程中有什么单词可以留下。由于是把字符串分为多个部分，故我们类比其他的分割字符串的问题，列出动态规划的方程如下：

$$f[l, k] = \max_{1 \leq i \leq l-1} \{ f[l-i, k-1] + g[l-i+1, i] \}$$

这里有初值  $f[l, 1]$  为：

$$f[l, 1] = g[1, l]$$

这个式子中， $f[l, k]$  表示把字符串前  $l$  个字符分成  $k$  段时所形成的最多单词的数目， $g[p, i]$  表示字符串的第  $p$  个字符开始的  $i$  个字符形成的字符串中所能形成的单词数。这里  $g[p, i]$  由于过于庞大，不能用预计算的方法加快速度，只能现用现算。计算的方法为对于所有  $p \leq q \leq p+i-1$  的  $q$ ，如果  $mlen[q]$  存在（也就是有单词可以在位置  $q$  匹配上），且  $q + mlen[q] - 1 \leq p+i-1$ ，则该位置必然可以匹配上一个单词。把所有这样位置的数目加起来就可以得到  $g[p, i]$  的值了。这样算法的复杂度为  $O(kl^3)$ 。

但这里计算还有一个技巧，就是我们可以依次按照  $k$  增加， $l$  增加， $i$  增加的顺序计算  $f$  的值。这样在  $i$  由  $i'$  增加到  $i'+1$  的时候，由于在计算  $i'+1$  所对应的  $g$  值时可以用

$$\begin{aligned} g[p-1, i'+1] &= g[p, i'] + 1 & p-1 + mlen[p-1] - 1 \leq p+i'-1 \\ &= g[p, i'] & p-1 + mlen[p-1] - 1 > p+i'-1 \end{aligned}$$

这个方程进行复杂度为  $O(1)$  的递推计算，故总复杂度可以降到  $O(kl^2 + wl^2)$ 。这种被称作双重动态规划的方法，请读者自己好好体会。

**数据：**

	输入	输出
1	2 1 thisisappleisthisthe	8

<sup>5</sup> 当然前提是在这个位置可以匹配上一个单词。

<sup>6</sup> 这里  $l$  为该字符串的长度。

<sup>7</sup> 这里  $w$  为单词的个数。

	oopbooktheisurrtoywe 4 is of the book	
2	4 4 dfhfgghgdfksgdfisdsds sdsdsddfsdffssddsfd asasassasdsdsdsdsdsd sadasadasdsdsdsdsd 4 dssd dfdfdf sdsd jkijk	13
3	10 4 aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaa 1 aaaaa	193
4	10 4 sdfdsassdasdddsasdd sdasdsasdsadasdasda assasaasadassadsaadd ssasdasdasdssdsassa asdasdasdasdasddsads dsadasdasdasadssdssa asssdasdsasdasdssaa dsaddsasdasdsadsaasa adsadsasddsadsadssa adsdsaasddsadsdsasa 6 aa sa	125

	ds da sa sd	
5	10 4 sdfdsassdasdddsasdd sdasddassdsaadaasdsa assasaasadassadsaadd ssasdsaasdassddsassa saddssassasdsaasssds dsasdasdsddasdasdssa asssdasdsasdasdssaa sadsassdssassdsasssa sasdsasdsasdsasdsssa sdsasdsasdsassdasdsa	65
	6 asd dsa ads das sad sda	

## 第四题：CAR 的旅行路线

又到暑假了，住在城市 A 的 Car 想和朋友一起去城市 B 旅游。她知道每个城市都有四个飞机场，分别位于一个矩形的四个顶点上，同一个城市的两个机场之间有一条笔直的高速铁路，第  $i$  个城市高速铁路的单位里程价格为  $T_i$ 。任意两个不同城市的机场之间均有航线，所有航线单位里程的价格均为  $t$ 。

那么 Car 应如何安排到 B 的路线才能尽可能的节省花费呢？她发现这并不是一个简单的问题，于是她来向你请教。

### 分析：

我们换一种对题目描述的方法，就是给出一张地图，求出某人从某点到另一点的最短距离。这是一个图论中的标准问题，就是在一个无向图中求最短路径的问题。

首先，这个人在从起点到终点所可能停下的点都是确定的，就是一个城市的四个机场（其他的时候是没有更换交通工具的自由）。所以我们可以以所有的机场为顶点，而机场与机场之间的交通路线<sup>8</sup>为边建立一个图。该图的边权值为机场之间相互通行所需的时间。

<sup>8</sup> 可能为铁路或航线。

至于求最短路，我们可以使用一个被称为 Dijkstra 的算法。该算法的主要思想就是模拟液体等速的在管子内流动，先流到的位置就是离起点较近的位置。在使用算法实现的时候，我们可以把顶点划分为已流到的和未流到的两个部分。依次找出液体从已流到的部分最少还需经过多少时间才能流到未流到的部分，并模拟流的过程。有关该算法的具体内容，请读者自行参见有关图论的算法书籍，这里不赘述。

最后，还需注意的是题目中对于一个城市只给出了三个机场的坐标。但我们知道这四个机场是成矩形排列的，而矩形的对角线互相平分。故我们可以先找出这三个机场中相对的两个机场，易知这样的机场就是距离最大的两个机场。然后通过对这两个机场求平均数求出该矩形的中心，再把另一个机场按矩形的中心作对称就可以得出第四个机场的坐标。还有题目中最多可能有 400 个节点，也就是说可能有 80000 条边。这些边显然是无法事先计算并保存下来的。但由于在求最短路径的过程中，每一条边最多只会访问两遍，故我们可以采用现用现算的办法。其他在计算点与点之间距离时也要注意对整数取平方时的运算有可能引发整数越界的问题，我们应该先转换成实数再进行计算。

该算法的时间复杂度为  $O\left(\frac{3}{2}n^2\right)$ ，空间复杂度为  $O(n)$ 。

数据：

	输入	输出
1	1 1 10 1 1 1 1 2 2 2 1 10	0.00
2	1 3 10 1 3 2 2 2 1 1 2 10 2 12 12 2 22 12 1 22 22 22 32 32 22 10	214.14
3	1 3 10 1 3 10 1 1 10 1 1 10 20 1 30 1 30 111 1 10 110 10 111 1 111 10	310.00
4	1 10 10 1 2 27 27 194 105 194 27 10 366 290 381 290 366 305 10 80 158 196 245 196 158 1 289 154 358 154 358 86 2 17 284 84 350 84 284 1 175 289 292 289 175 362 3 450 371 420 371 420 32 2	1885.03



	241 29 270 29 270 43 4	
	251 182 347 270 251 270 5	
	347 341 410 341 347 369 1	
	1	
	15 50 1 15	
	1 2 3 1 4 3 7	
	1 12 3 11 4 13 36	
	1 22 3 21 4 23 5	
	1 32 3 31 4 33 19	
	1 42 3 41 4 43 47	
	11 2 13 1 14 3 2	
5	11 12 13 11 14 13 6	1924.23
	11 22 13 21 14 23 1	
	11 32 13 31 14 33 27	
	11 42 13 41 14 43 29	
	21 2 23 1 24 3 34	
	21 12 23 11 24 13 16	
	21 22 23 21 24 23 14	
	21 32 23 31 24 33 41	
	21 42 23 41 24 43 3	

关于 yuwei 和 zhaoshuang 的报告的一点补充

第一题：其实最简单的方法是这样子的：让  $x$  从  $-100.00$  到  $100.00$  枚举一下，得到 20000 个  $f(x)$ ，取跟 0 最接近的三个  $f(x)$ ，对应的  $x$  就是答案了。（提示有时候会扰乱别人的思维）

第二题：如果时间不是很严格的话，枚举还是能过的。

第三题：这跟我在分区赛前出的一道模拟试题方法类似。不过分区这题的数据巨弱，居然只输出“有多少个位置开始至少可以有一个单词”也能对 4/5 的点！真是佩服出数据的人……

第四题：标准算法的题目，不过这题算老题目了……算费用的时候，令 A,B 城市内的路费为 0 的话，就可以直接得到结果，而不需要做 4 遍 Dijkstra。当然，用 Flody 也可以做