

一题多解 探析二分

冯彦伟

浙江省义乌大成中学

DOI:10.12238/mef.v4i8.3996

[摘要] “二分算法”在浙江选考中频频出现，成为了一线信息教育老师教学中的一个重点内容。在“二分查找”算法学习过程中，大部分学生对于查找区间、中间值比较、对半查找这些字眼都很熟悉，讲述算法思想也能八九不离十，但遇到题目变式就是不能举一反三。针对这一问题，本文对“二分算法”各种变式采用一题多解进行分析，探讨不同理解下的程序实现。

[关键词] 二分算法；举一反三；一题多解

中图分类号：G632

文献标识码：A

One Question with Multiple Solutions and Analysis of Dichotomy

FENG Yanwei

Yiwu Dacheng Middle School

[Abstract] "Dichotomous algorithm" appears frequently in Zhejiang election examination, and has also become a key content in the teaching of front-line information teachers. In the process of learning the "binary search" algorithm, most students are familiar with the words of search interval, intermediate value comparison and half search. They can also tell the idea of the algorithm, but they can't draw inferences from one example when encountering problem variants. Aiming at this problem, this paper analyzes various variants of "dichotomous algorithm" by using one problem and multiple solutions, and discusses the program implementation under different understanding.

[Key words] dichotomous algorithm; draw inferences from one example; one question with multiple solutions

【题干】 为了让学生适应赋分，同时更好地鼓励学生，我们定义了一个班级赋分表。每个分数固定相对应的人数，如下表所示：

班级共有40人，每个分数段的人数存在a数组，每个学生的成绩从数据库中读取存在c数组，并已经降序排序。现请设计一个程序，可供学生查询自己的赋分。

例如前10个学生技术成绩依次为：

82、82、81、81、78、76、76、74、71、70

按照题目要求进行赋分，括号内为赋分分数：

82 (100)、82 (100)、81 (97)、81 (97)、78 (94)、76 (94)、76 (94)、

分数	100	97	94	91	88	85	82	79	76	73	70	67	64	61	58
人数	1	3	3	3	3	3	3	4	3	2	2	3	3	3	1

74 (91)、71 (91)、70 (91)

解法1：熟练基础代码，巩固基础模型

【解题思路】 根据情境，c数组已经将技术成绩降序存储，赋分又是根据人数进行，那么确定输入分数key的名次就是解题关键。先对c数组进行二分查找，确定key的位置，这里有一个问题需要注意，即有同分的情况出现，而确定名次需要找到第一个key所在的位置。所以，我们可以在找到的位置基础上，继续往前查找，找到前一个不等于key结束，就能得出key的名次。根据名

次和赋分人数a数组，既可以得到对应的赋分。

【参考核心代码】

```
Dim a(1 To 15) As Integer, c(41) As Integer
Dim i As Integer, j As Integer, key As Integer, m As Integer
' 初始化a数组
a(1) = 1: a(2) = 4: a(3) = 7: a(4) = 10: a(5) = 13: a(6) = 16
a(7) = 19: a(8) = 23: a(9) = 26: a(10) = 28: a(11) = 30
a(12) = 33: a(13) = 36: a(14) =
```

39: a(15) = 40

'数据库读取技术成绩存储在c数据, 并降序排序, 代码略

key = Val(Text1.Text)

```
i = 1: j = 40      '初始化查找区间
Do While i <= j    '查找 key 所在位置
    m = (i + j) \ 2
    If key < c(m) Then
        i = m + 1
    ElseIf key > c(m) Then
        j = m - 1
    Else
        Exit Do
    End If
Loop
Do While a(m - 1) = key    '查找第一个 key 所在位置
    m = m - 1
Loop
```

i = 1

Do While a(i) < m'

确定赋分

i = i + 1

Loop

Text2.Text = Str(100 - 3 * (i - 1))

将框中的代码进行思考后, 考虑能否合并?

分析: 此题目的要查找第一个key所在的位置, 那么找到key时可以分为两种情况讨论, 前一个还是key, 前一个不是key。即用嵌套选择结构实现。所以框中部分代码可以修改为:

i = 1: j = 40

初始化查找区间

Do While i <= j

查找第一个key所在位置

m = (i + j) \ 2

If key < c(m) Then

i = m + 1

ElseIf key > c(m) Then

j = m - 1

Else

If c(m - 1) = key Then j

= m - 1 Else Exit Do

End If

Loop

思考: (1) 为什么 i=1, j=40? (2) 为什么循环条件 i<=j? 为什么

i=m+1, j=m-1?

(3) m=(i+j)\2 和 m=(i+j+1)\2 有什么区别?

解法2: 理解算法核心, 深化思维层次

【解题思路】经过上面的思考, 我们不难得出在二分查找时, i、j代表我们查询的区间, 确保查找对象在这个区间内有可能出现, 而“i=m+1”, “j=m-1”, 是因为m的位置我们已经比较过, 不是我们要的答案。

根据上面的分析, 我们在查找第一个key时, 我们找到了key是想往前继续查找, 但纠结万一当前的key是第一个key怎么办, 如果继续“j=m-1”, 那么不就错过了第一个key。结合考虑, 既能使查找区间往前, 又能保证第一个key在查找区间内, 我们可以用“j=m”。但是其他结构是否要相应改变?

基础模型中 i<=j 的分析:

i、j分别指向查找区间, i是由 i=m+1 得到, j是由 j=m-1 得到, i、j指向的位置都未与key比较, 所以当 i=j 时, 这个位置还未与key比较, 理应再比较一次。

而当 i 是由“i=m+1”得到, j 是由“j=m”得到, j指向的位置是已经与key比较过, 所以当“i=j”时, 没有必要再去key比较。

【参考核心代码】

Dim a(1 To 15) As Integer, c(41)

As Integer

Dim i As Integer, j As Integer,

key As Integer, m As Integer

' 初始化a数组

a(1) = 1: a(2) = 4: a(3) = 7: a(4)

= 10: a(5) = 13: a(6) = 16

a(7) = 19: a(8) = 23: a(9) = 26:

a(10) = 28: a(11) = 30

a(12) = 33: a(13) = 36: a(14) =

39: a(15) = 40

' 数据库读取技术成绩存储在c数据, 并降序排序, 代码略

key = Val(Text1.Text)

i = 1: j = 41

' 初

始化查找区间

Do While i < j

' 查找

第一个key所在位置

m = (i + j) \ 2

If key < c(m) Then

i = m + 1

Else

j = m

End If

Loop

j = 1

Do While a(j) < i

' 确定赋分

j = j + 1

Loop

Text2.Text = Str(100 - 3 * (j - 1))

思考: 这里如果将“m=(i+1)\2”改成“m=(i+j+1)\2”, 是否可行? 为什么?

当查找区间剩下两个数时, 如果“m=(i+j+1)\2”, 那么“m=j”。如果“m=(i+j)\2”, 那么“m=i”。

解法3: 拓宽思维广度, 敢于尝试

既然解法2能够将两种分类情况合并利用“j=m”实现, 那我们能不能再进行一个大胆的尝试, 在原先基础模型的基础上, 找到key时不管c(m-1)是否等于key都直接往前找, 运行“j=m-1”。

【解题思路】根据之前我们理解, i、j代表二分查找区间, 若直接运行“j=m-1”, 那么我们可能导致c(i)-c(j)之间没有key, 好像是错了。接下来我们通过两种情况分析, 来讨论是否可行。

(1) 若c(m)此时已经是第一个key, 运行条件j=m-1, 那么新区间c(i)-c(j)都大于key, 即都运行“i=m+1”, 直到i>j, 即i在上图中m的位置。

(2) 若c(m)不是第一个key, 那么运行“j=m-1”, 第一个key还是在c(i)-c(j)区间, 最终还是会运行到某个m' 是第一个key, 重复(1)操作。

【参考核心代码】

```

Dim a(1 To 15) As Integer, c(41)
As Integer
Dim i As Integer, j As Integer,
key As Integer, m As Integer
' 初始化a数组
a(1) = 1: a(2) = 4: a(3) = 7: a(4)
= 10: a(5) = 13: a(6) = 16
a(7) = 19: a(8) = 23: a(9) = 26:
a(10) = 28: a(11) = 30
a(12) = 33: a(13) = 36: a(14) =
39: a(15) = 40
' 数据库读取技术成绩存储在c数
据, 并降序排序, 代码略
key = Val(Text1.Text)
i = 1: j = 40 ' 初始
化查找区间
Do While i <= j ' 查
找第一个key所在位置
m = (i + j) \ 2
If key < c(m) Then
i = m + 1
Else

```

```

j = m - 1
End If
Loop
j = 1
Do While a(j) < i
' 确定赋分
j = j + 1
Loop
Text2.Text = Str(100 - 3 * (j -
1))

```

思考: 依据题目中二分查找自身的语句, 去分类讨论证明查找结束时, i、j的位置所表达的意义, 只需要掌握其中方法, 就可以避免死记硬背。在本文提到的案例情境中, 三个解法最终确定赋分都是利用顺序查找a数组来实现的, 大家可以动手尝试一下, 将顺序查找改为二分查找。

结语

只有教师真正帮助学生稳住“举一”, 学生才有可能“反三”, 而不是空谈学生不会迁移。所以个人任务, 算法

课的教学, 前期并不需要多复杂的例子, 而是“复杂”短短几行代码, 从中挖掘出更多的信息, 提供更多的时间让学生思考, 嚼烂再精, 而不是来回炒冷饭。当然算法模型建立, 基础的理解是底层建筑, 必须牢固, 本文中几种解法的分析和证明方法既是提供一个巩固二分基础模型的思路。

[参考文献]

[1]任宏梅.《二分查找算法实现》教学设计[J].中国信息技术教育,2019(07):32-35.

[2]张河炳.二分查找算法的运用策略研究[J].福建电脑,2018,34(8):116-117.

[3]石岩.浅谈高中数学“一题多解”教学中如何培养数学创新思维[J].考试周刊,2021(61):79-81.

作者简介:

冯彦伟(1995—),男,汉族,浙江义乌人,中级二级,本科,从事中学信息教育工作。

