

# 信息技术选考专题

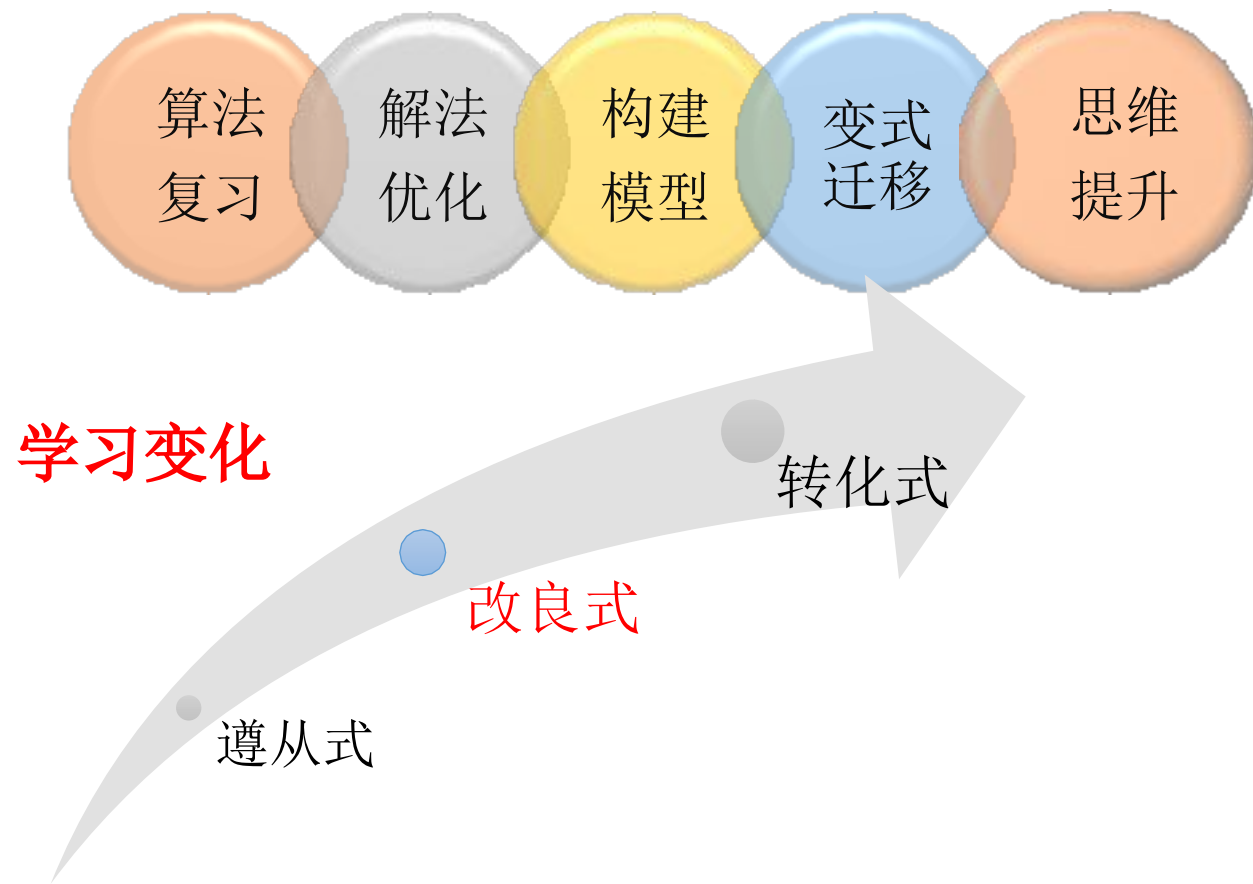
浙江省杭州第十一中学

刘正阳

# 专题1：二分查找

## 单元设计思路：

1. 传统算法在key值未知时分析繁杂，效率偏低；
2. 优化解法，构建二分查找判定树模型并探究规律
3. 判定树用于解决该类问题及拓展，提高效率。



# 算法复习

1) 已知待查找元素，使用表格法跟踪运行过程；

例题1: 有如下VB程序段：

```
i = 1 : j = 10 : x=18
Do While i <= j
    m = Int((i + j) / 2)
    If x = f(m) Then Exit Do
    If x < f(m) Then j = m - 1 Else i = m + 1
Loop
```

f(1)到f(10)的值依次为：2, 7, 8, 10, 12, 13, 16, 18, 19, 20,  
该段程序运行后，正确的是 ( C )

A.  $i=m+1$     B.  $j=m-1$     C.  $j>m+1$     D.  $i>m-1$

| 初始值 | i=1 | j=10 | m | 过程         |
|-----|-----|------|---|------------|
| 第一趟 | 6   |      | 5 | $x > f(m)$ |
| 第二趟 |     |      | 8 | 找到         |

退出循环体时，i, j, m 的值???

6, 10, 8

## 二叉查找判定树的引入：

例2【2019.1金丽衢三市联考】有如下程序段：

```
i=1: j=10: n=0: flag=true
```

```
Key=Val(text1.text)
```

```
Do While i<=j and flag=true
```

```
  m=(i+j)\2
```

```
  if a(m)=key then
```

```
    flag=false
```

```
  elseif key>a(m) then
```

```
    i=m+1: n=n-1
```

```
  else
```

```
    j=m-1: n=n+1
```

```
  end if
```

```
Loop
```

数组元素a(1) 到 a(10)的值依次是：

“5, 16, 22, 28, 35, 43, 52, 67, 78, 89”，变量n的值最终是0，则文本框Text1输入的数值范围可能是（ B ）

A. (28, 35)

B. (43, 52)

C. [52, 67]

D. [78, 89]

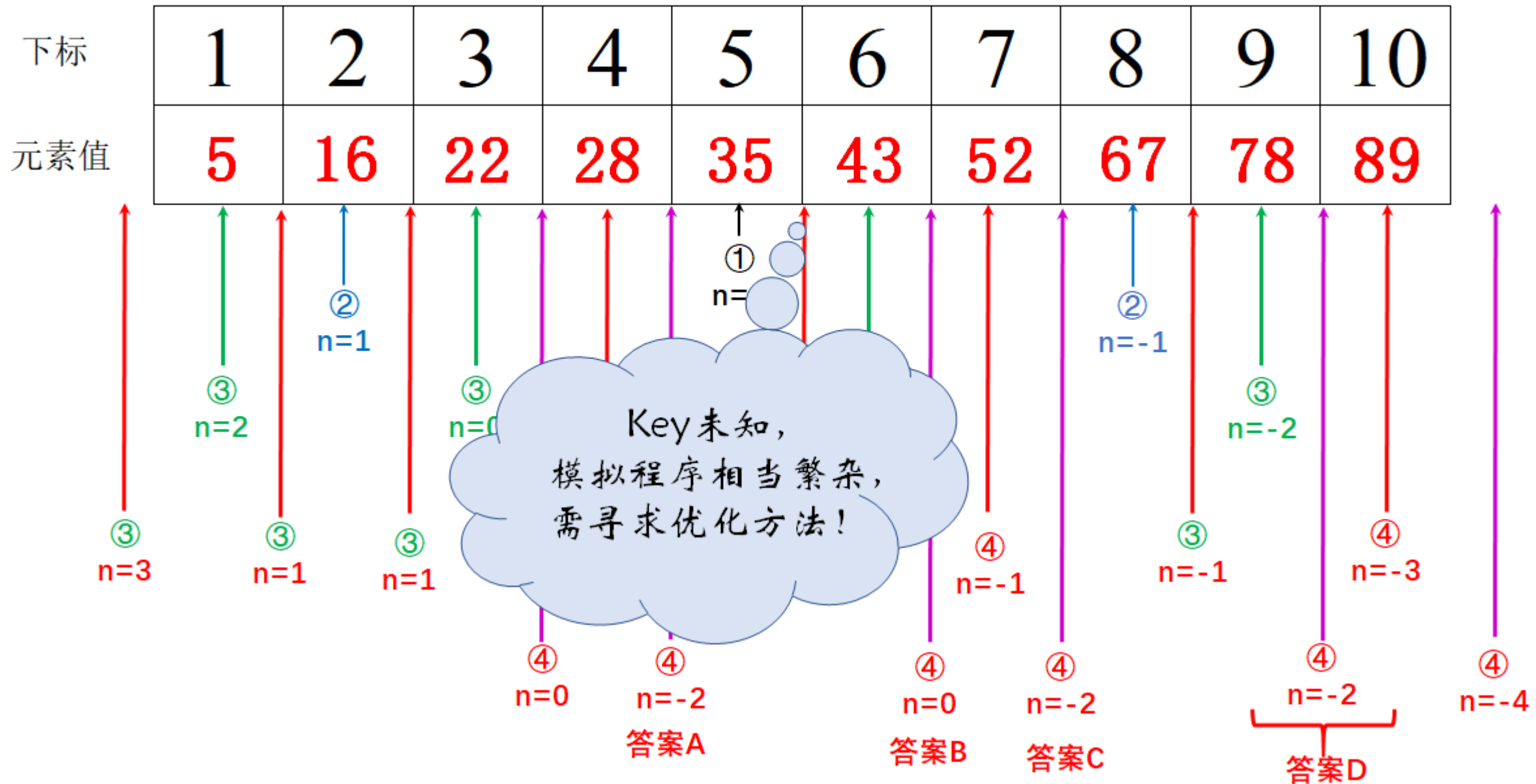
传统分析法：

法1：特值法，不足：难以用于填空形式；

法2：需要模拟程序，列表分析逐m逐区间求解出n并判定

# 思维始于疑难

传统列表分析法：

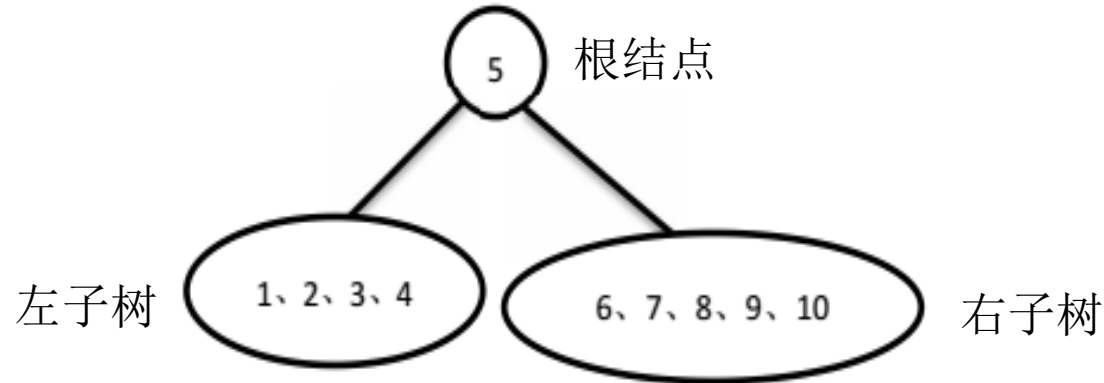


# 构建模型

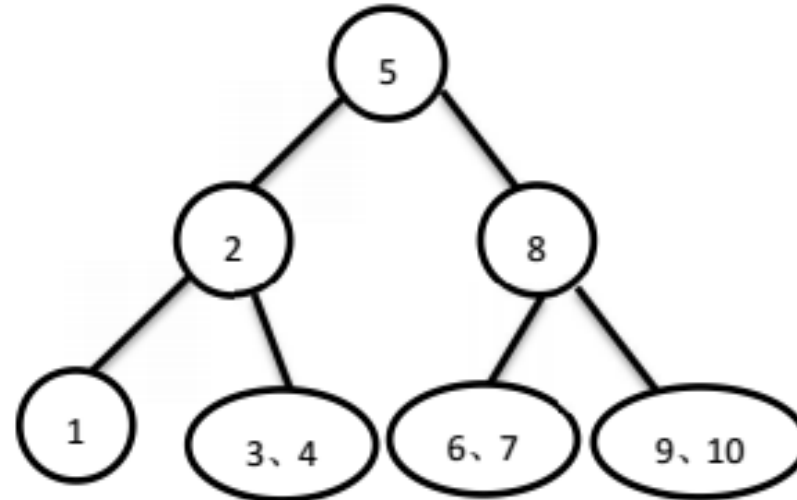
## 板演：二分查找判定树的构建（以数组元素升序为例）

假设先有10个数据1, 2, 3, 4, 5, 6, 7, 8, 9, 10

①把当前查找区间的中间位置上的结点（即m值）作为根，左子数组和右子数组中的结点分别作为根的左子树和右子树。



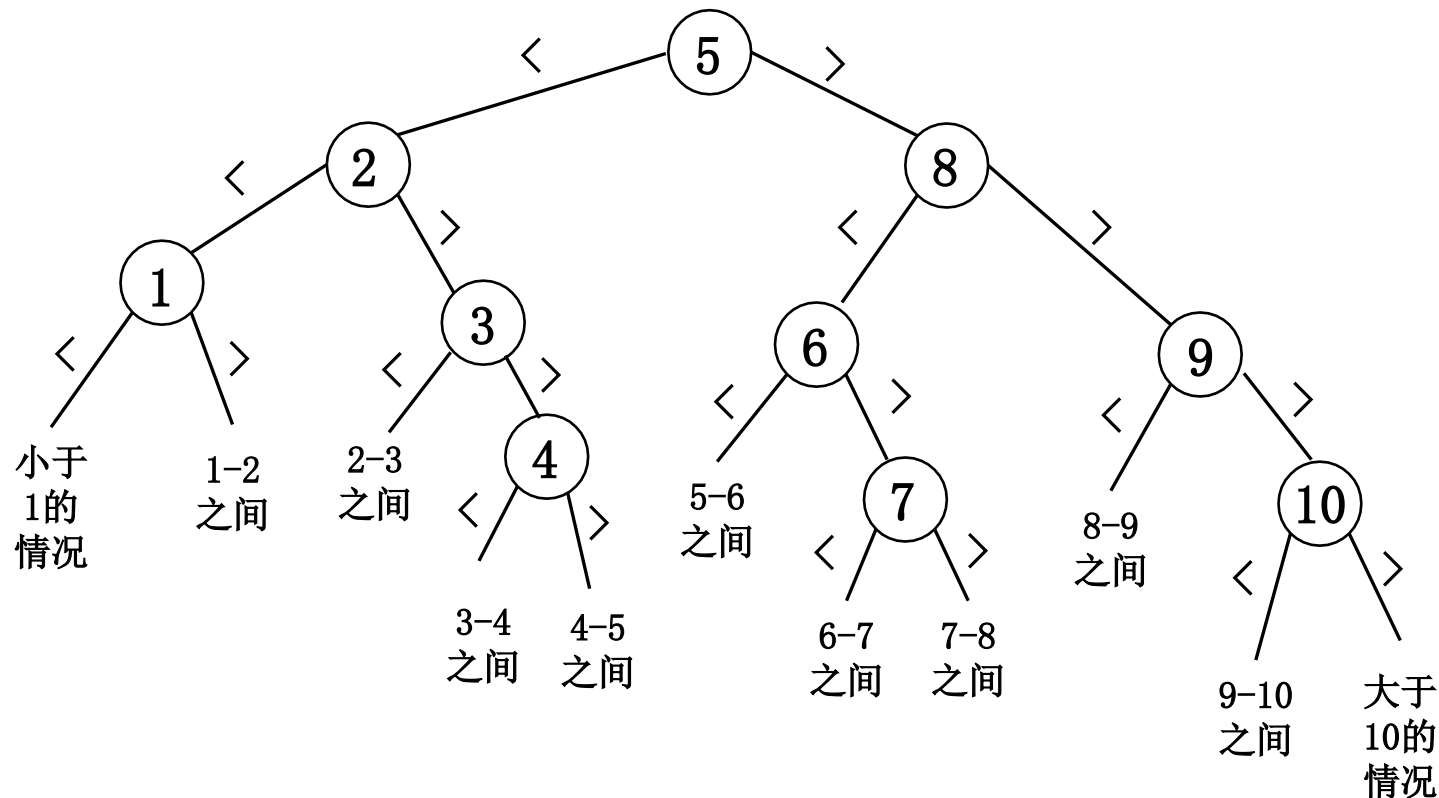
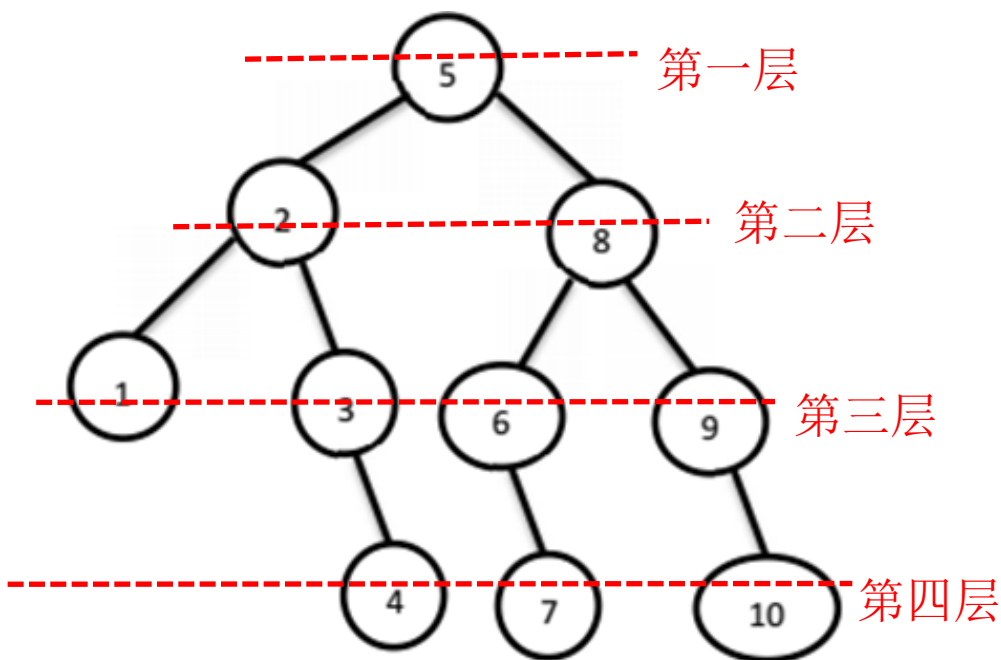
②左右两堆依次求出m值（2、8），m值保留在原位，然后把两边的数分别放入它的左右两个子树（小的放左子树，大的放右子树）：



原始数据 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

③ 结点里还有2个及以上的数的，按照上面规则求m值，m值保留在原位，其它数放入它的左右两个子树（小的放左子树，大的放右子树）：

④ 没有左子树的往左画条线，代表往左查找失败的范围；没有右子树的往右画条线，代表往右查找失败的范围。



## 二叉查找判定树的应用:

# 繁简、效率对比

演练【例2的二分查找判定树解法】有如下程序段:

```
i=1: j=10: n=0: flag=true
```

```
Key=Val(text1.text)
```

```
Do While i<=j and flag=true
```

```
  m=(i+j)\2
```

```
  if a(m)=key then
```

```
    flag=false
```

```
  elseif key>a(m) then
```

```
    i=m+1: n=n-1
```

```
  else
```

```
    j=m-1: n=n+1
```

```
  end if
```

```
Loop
```

数组元素a(1) 到 a(10)的值依次是:

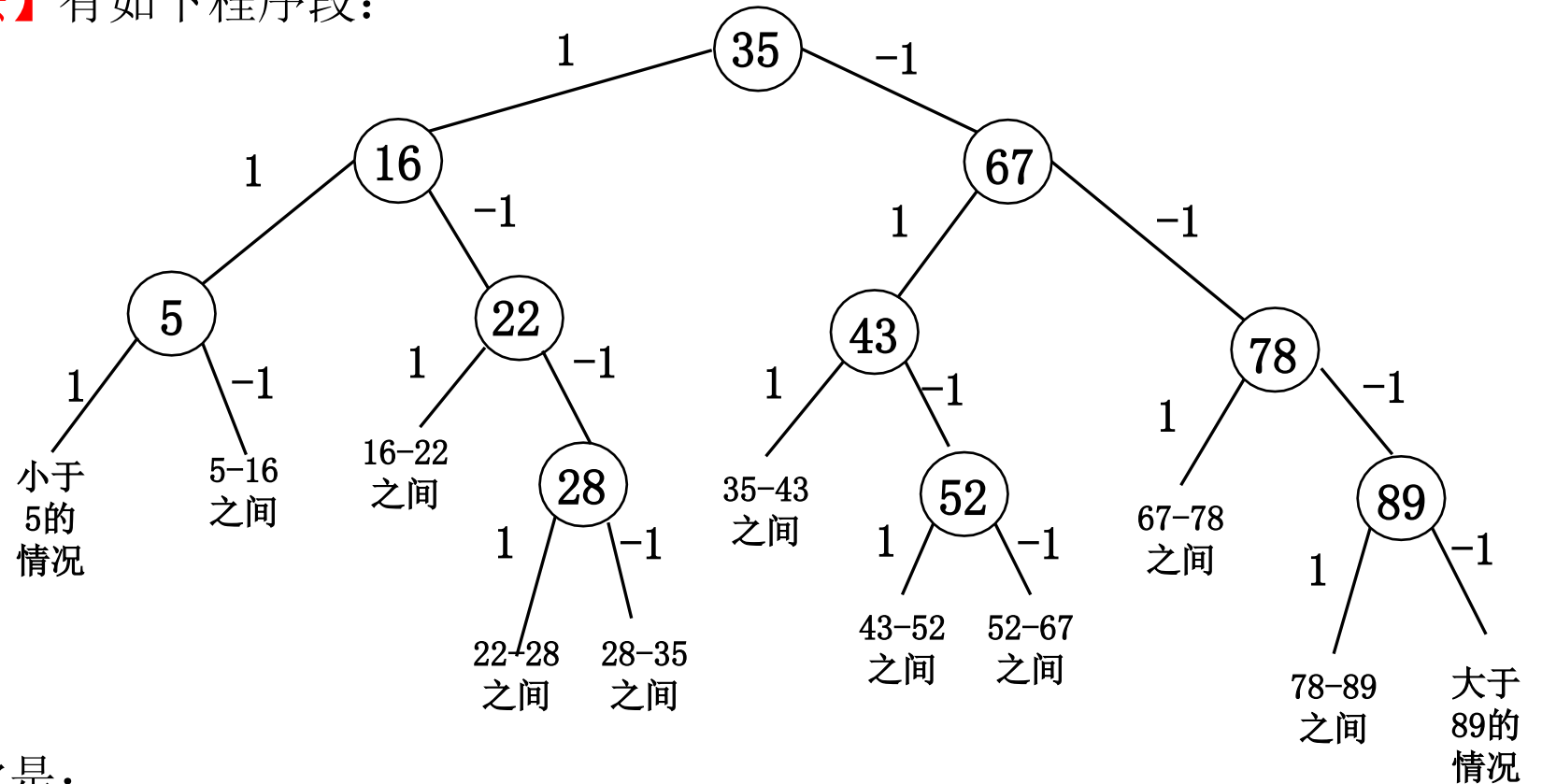
“5, 16, 22, 28, 35, 43, 52, 67, 78, 89”, 变量n的值最终是0, 则文本框Text1输入的数值范围可能是 ( **B** )

A. (28, 35)

B. (43, 52)

C. [52, 67]

D. [78, 89]





## 规律小结1： 看黑板演示图，找规律

①关于查找次数 $c$ 和判定树层数关系：

例2中：元素22在第3层，需查找3次；

元素52在第4层，需查找4次；

若查找值为20的某元素，则从结点22（第3层）左边走出，需查找3次；

若查找值为25的某元素，则从结点28（第4层）左边走出，需查找4次；

若查找值为55的某元素，则从结点52（第4层）右边走出，需查找4次；

因此找到时，查找次数就是该数（结点）所在的层数；（查找次数和层数关系）

若找不到时，查找次数就是最后走出结点所在的层数；（查找次数和最后走出结点层数关系）

# 变式迁移：问法变化

小试牛刀1【2017.11浙江】某对分查找算法的VB程序段如下：

```
i = 1: j = 7: s = ""
```

```
key = Int(Rnd * 100)
```

```
Do While i <= j
```

```
    m = (i + j) \ 2
```

```
    If key = a(m) Then
```

```
        s = s + "M": Exit Do 'Exit Do 表示退出循环
```

```
    ElseIf key < a(m) Then
```

```
        j = m - 1: s = s + "L"
```

```
    Else
```

```
        i = m + 1: s = s + "R"
```

```
    End If
```

```
Loop
```

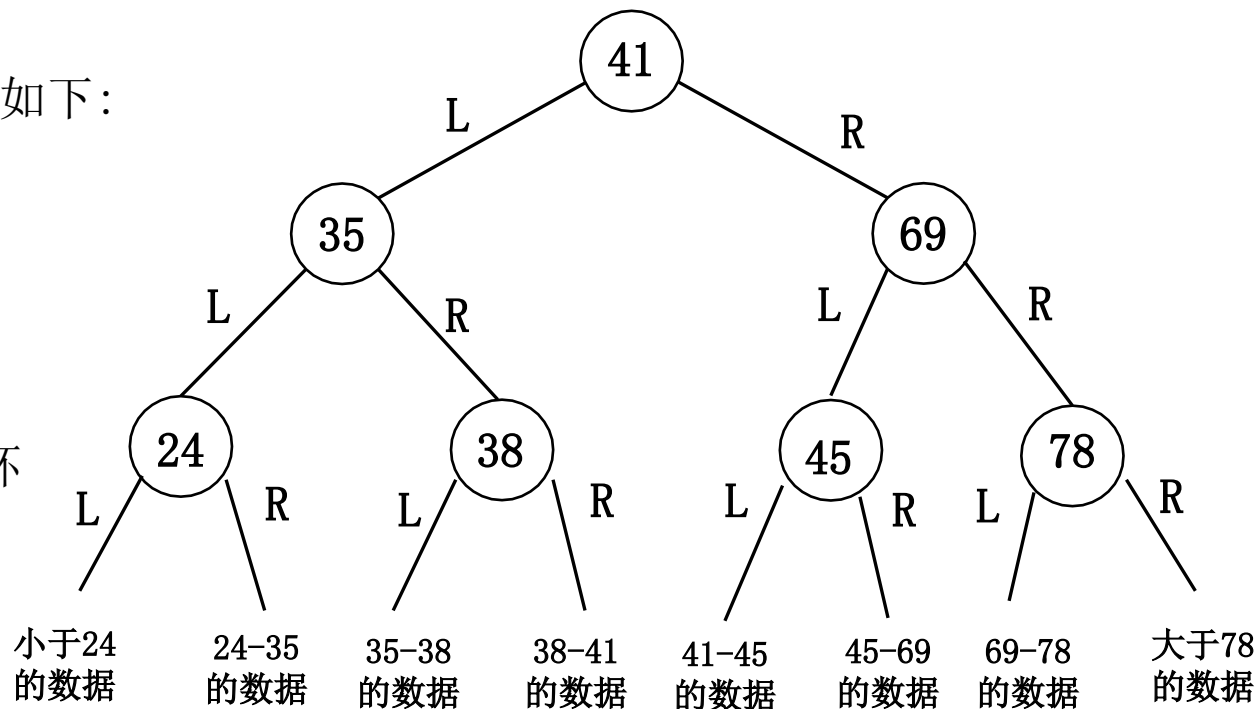
```
Text1.Text = s
```

1. 数组元素a(1)到a(7)的值为：“24, 35, 38, 41, 45, 69, 78”。

若该程序段执行后，文本框Text1中显示的内容可能是 ( )

A. RL                      B. LMR                      C. RLR

D. LRLM



解析：由二叉判定树可知，

A项“RL”已走到数据45，是能找到，必为“RLM”

B项“LM”即可找到，退出循环，不可能再有R；  
D项“LRL”走到了35-38之间找不到的数，其后不可能有“M”；

C项“RLR”代表输入的是45-69之间找不到的数。

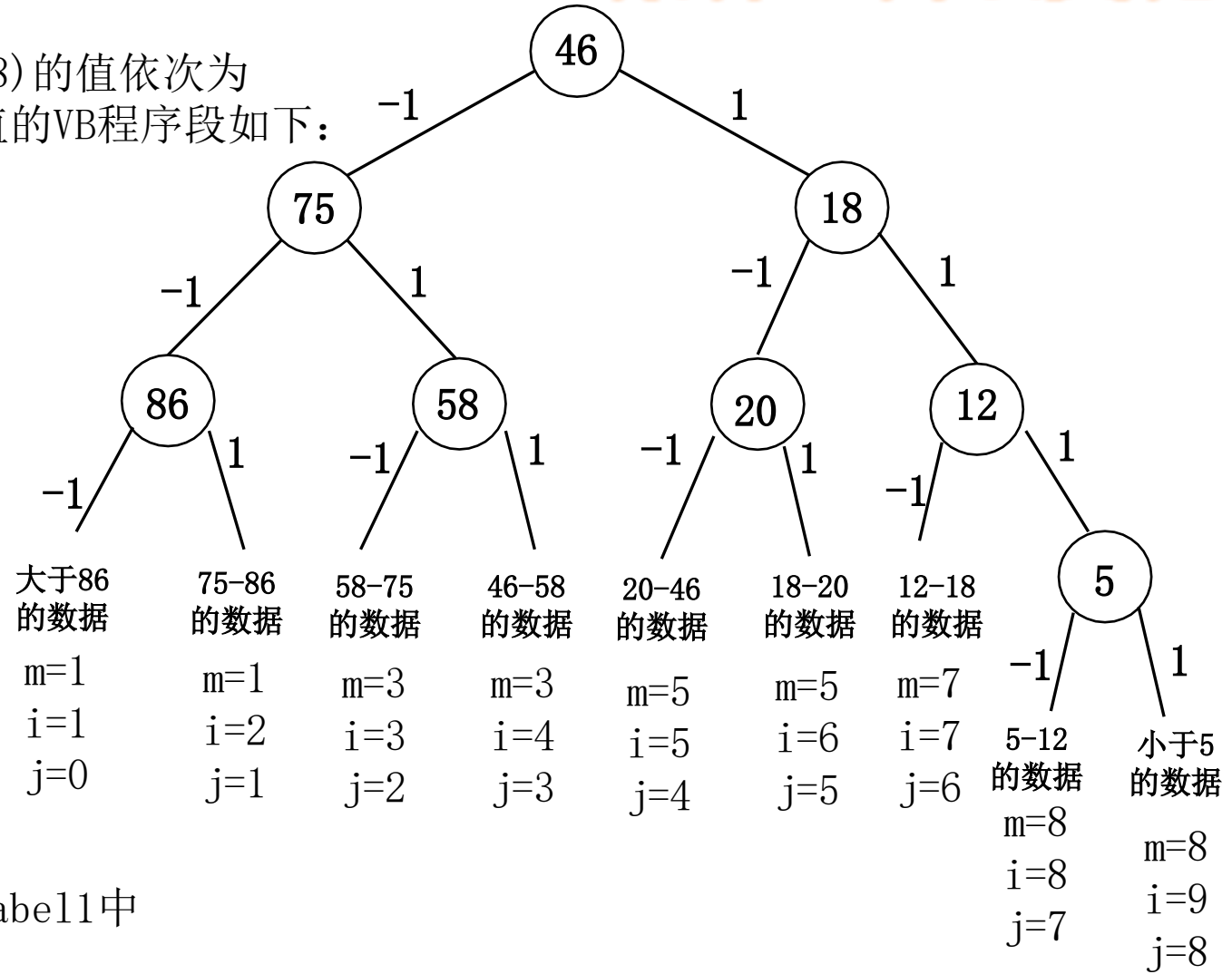
## 二叉查找判定树的应用:

## 条件、问法变化

小试牛刀2【2018.6杭州统测】若数组元素d(1)到d(8)的值依次为“86, 75, 58, 46, 20, 18, 12, 5”，查找某Key值的VB程序段如下：

```

n = 0 : i = 1 : j = 8
Key = Val(Text1.Text)
Do While i <= j
    m = (i + j) \ 2
    If Key = d(m) Then Exit
    If Key > d(m) Then
        j = m - 1 : n = n + 1
    Else
        i = m + 1 : n = n + 1
    End If
Loop
Label1.Caption = Str(n)
    
```



1. 当输入不同的Key值，运行该程序段后，在标签Label1中显示的不同结果共有 ( D )

- A. 5种      B. 6种      C. 7种      D. 8种

-3到4，共8种

2. 请分别写出当Key=50、Key=10情况下的m, i, j值。

Key=50时, m=3, i=4, j=3

Key=10时, m=8, i=8, j=7

## 规律小结2:

② 如果查找的数据找不到，得出*i*和*j*的值：

首先得出最后走出数（结点）的*m*，*i*，*j*值；

如果往左走出（ $\text{key} < a(m)$ ），则  $j = \underline{m-1}$ ， $i = \underline{m}$ ；

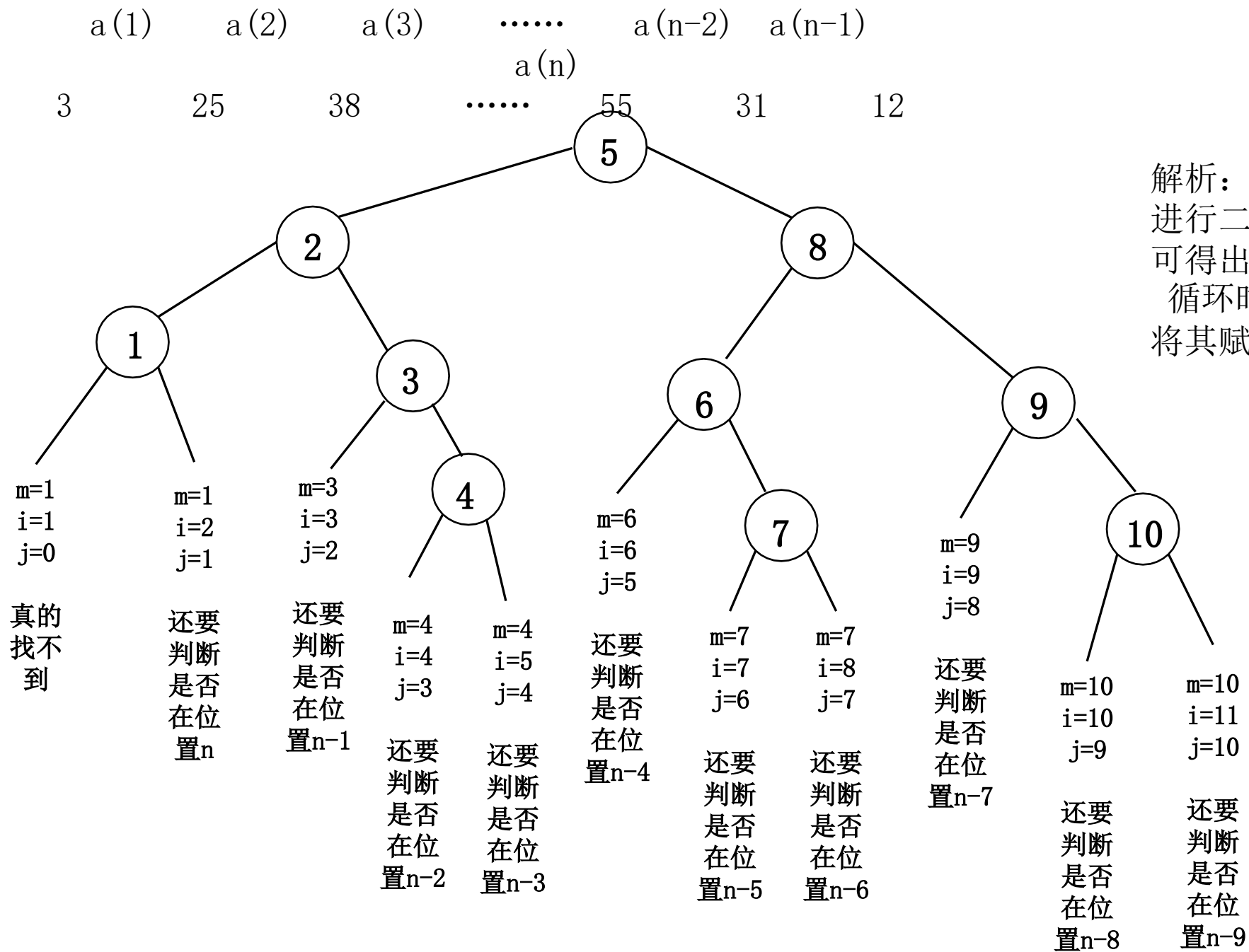
如果往右走出（ $\text{key} > a(m)$ ），则  $i = \underline{m+1}$ ， $j = \underline{m}$ ；

③ 如果有*n*个数字，找不到的区间共有  $\underline{n+1}$  个，并且有：

*i*依次从1、2……*n*+1，*j*依次从0、1……、*n*。





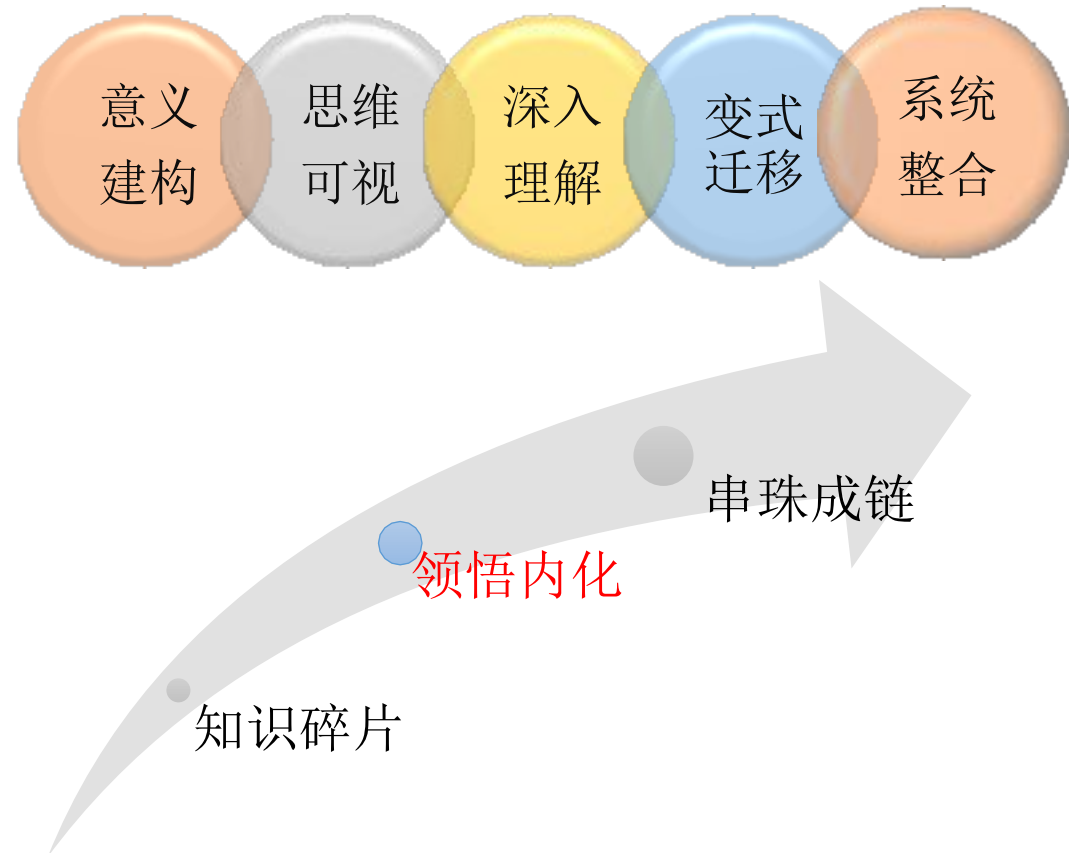
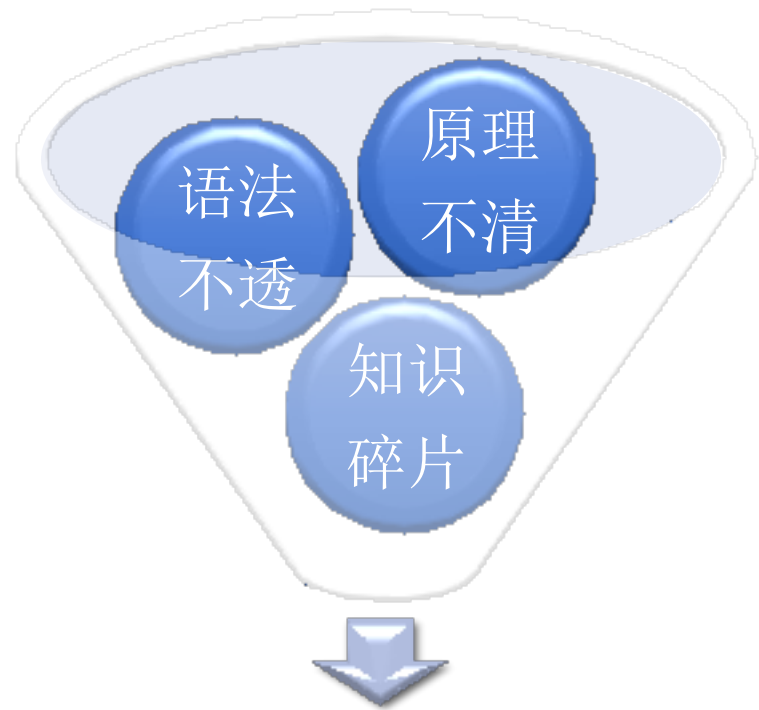


解析：假设  $n=20$ ，对左边一半进行二分查找，画出查找判定树，可得出左半查找不到时退出 Do 循环时的  $m$  对应到右半部分位置，将其赋值给新的  $m$  即可。

## 专题2：插入排序

### 单元设计思路：

1. 让学生亲历将实际问题抽象成结构模型；
2. 将结构模型分解，主动学习，各个击破；
3. 深入理解并内化，让学习在课堂真正发生。





**实际问题模型：**每次从桌子上拿走一张牌并将它插入到左手中正确的位置

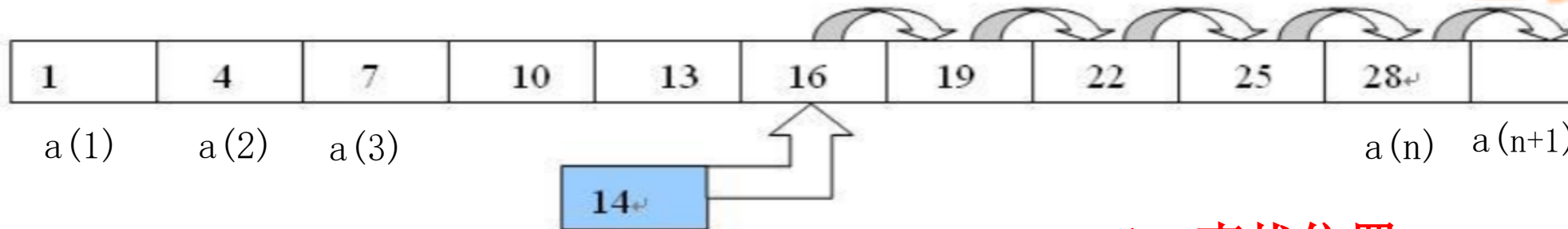


**意义建构：**其建构的意义是指事物的性质，规律以及事物之间的内在联系。

先研究插入**一个数据**： 重要的前提：手里的牌已经是排好序的。（约定：若数据相同则**顺接**插入）

# 意义建构

抽象模型：



## step1. 查找位置

# 问题分解

1. 从前往后查找位置：

```
For i=1 to n
  if key < a(i) then exit for
Next i
wz=i
```

```
i=1
Do While key >= a(i) and i <= n
  i=i+1
Loop
wz=i
```

2. 从后往前查找位置：

```
For i=n to 1 step -1
  if key > a(i) then exit for
Next i
wz=i+1
```

```
i=n
Do While key <= a(i) and i >= 1
  i=i-1
Loop
wz=i+1
```

3. 二分查找定位：（用于数据量大）

有重复数据，可用对分并看后一位，或微增量近似替代查找

插入排序  
问题分解

- step1. 查找位置
- step2. 移位腾出
- step3. 数据插入

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/127045023031006145>