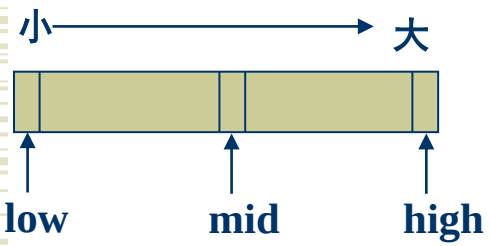
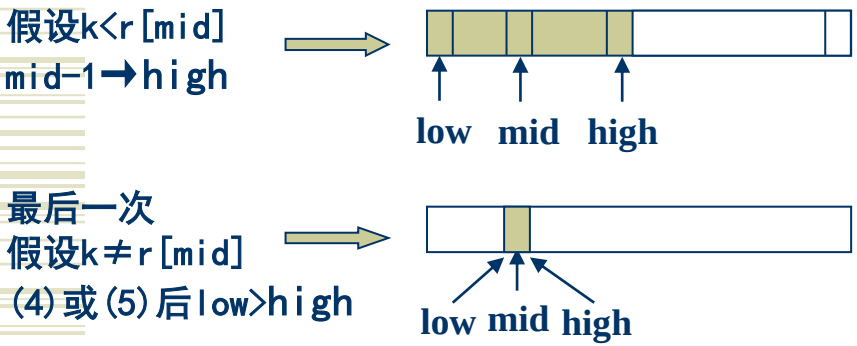
例子5.9：有一个从小到大顺序排列的无符号数数组，DI=数组首地址，数组中第一个单元存放数组长度，AX中有一个无符号数。要求在数组中查找与AX相等的数，如找到，则使CF=0，并在SI中给出该元素在数组中的相对位置；如未找到，则使CF=1，SI给出最后一个比较元素的偏移地址。

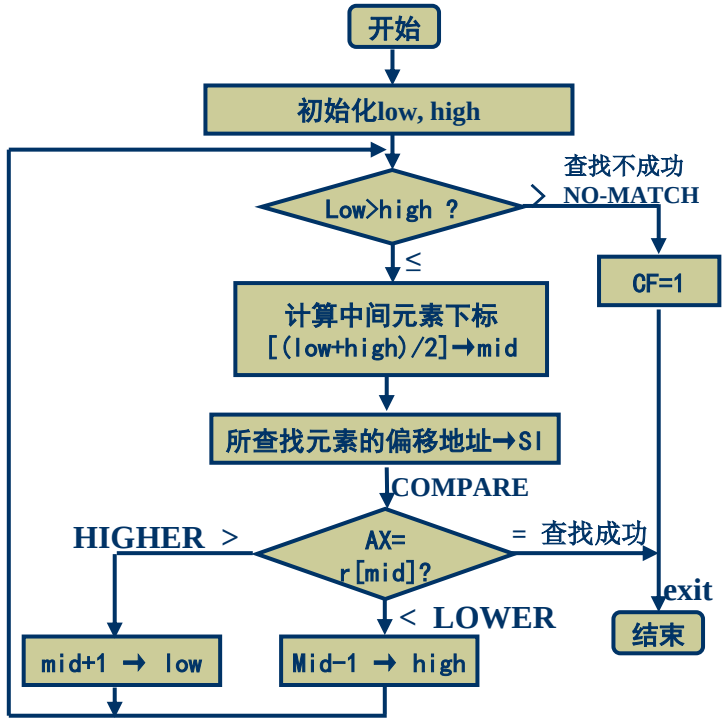
- 查找时：
  - 如果数据大小排序不定，只能用顺序查找方法；
  - 如果数据大小排序规整，也可用折半查找法，以提高查找效率

折半查找算法

在一个长度为n的有序数组r中，查找元素k的折半查找算法如下：
(1)初始化被查找数组的首尾下标：1→low, n→high;
(2)若low>high，则查找失败，置CF=1，退出程序。
　否则，计算中点：(low+high)/2 → mid;
(3)k与中点元素r[mid]比较。若k=r[mid]，则查找成功，程序结束；若k<r[mid]，
　则转步骤(4);若k>r[mid]，则转步骤(5); (假设数据由小到大排列)
(4)低半部分查找(lower)，mid-1 → high，返回步骤(2)，继续查找；
(5)高半部分查找(higher)，mid+1 → low，返回步骤(2)，继续查找；



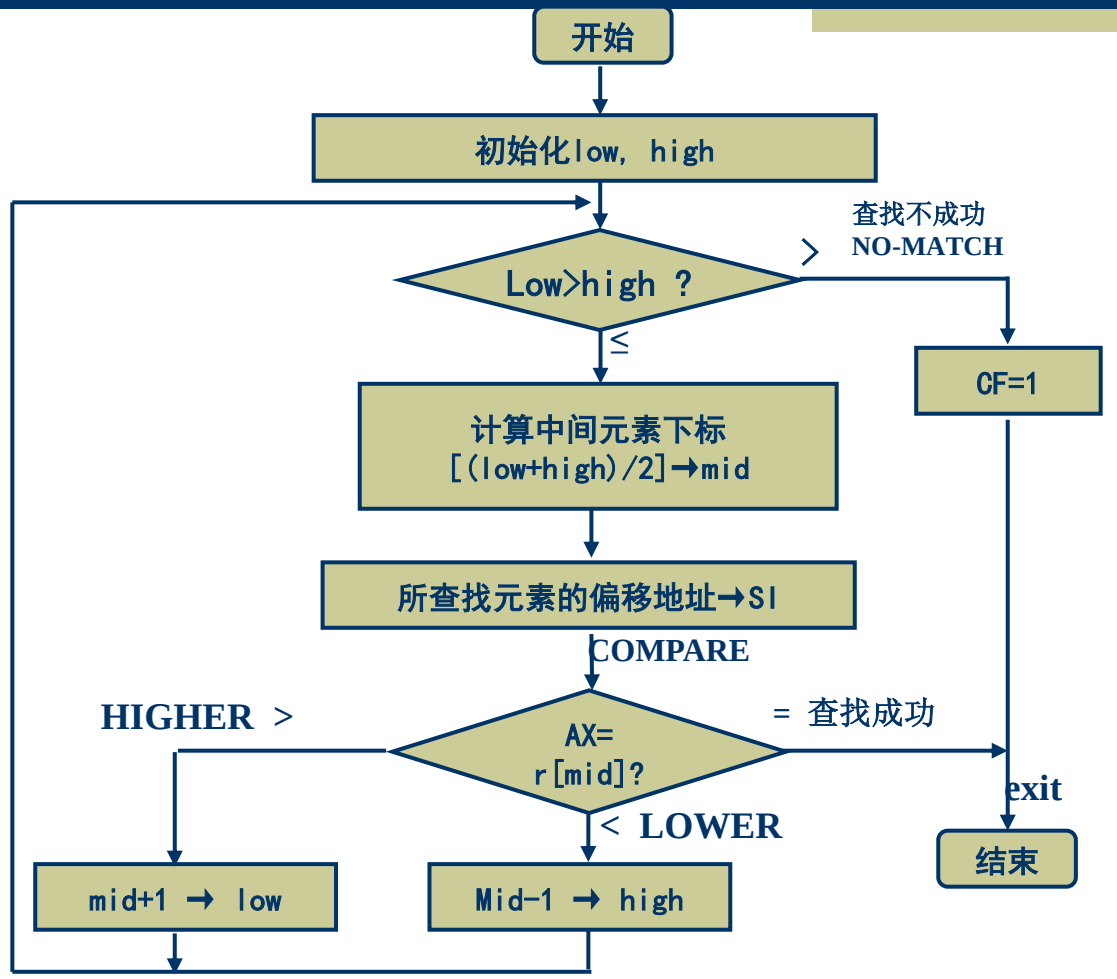




折半查找法

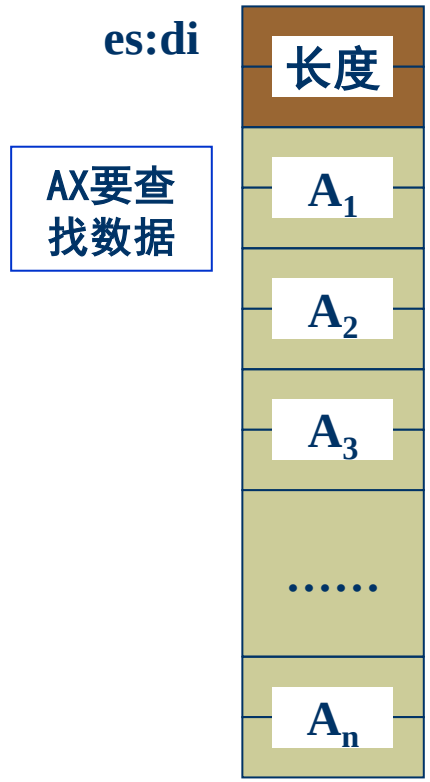
流程：图5.11
(参看核心程序段p194search～no_match)

参看p192.asm



```
dseg        segment
    low_idx         dw  ?
    high_idx        dw  ?
dseg        ends
cseg        segment
b_search  proc      near
    assume  cs:cseg
    assume  ds:dseg, es:dseg
    push    ds
    push    ax
    mov     ax, dseg
    mov     ds, ax
    mov     es, ax
    pop     ax

    cmp     ax, es:[di+2]
    ja      chk_last        ;ax>A1
    mov     si, 2
    je      exit            ;ax=A1
    jmp     no_match        ;ax<A1
chk_last:
    mov     si, es:[di]
    shl     si, 1
    add     si, di
    cmp     ax, es:[si]
    jb      search          ;ax<An
    je      exit            ;ax=An
    jmp     no_match        ;ax>An
search:
    mov     low_idx, 1
    mov     bx, es:[di]
    mov     high_idx, bx
    mov     bx, di
mid:
    mov     cx, low_idx
    mov     dx, high_idx
    cmp     cx, dx
    ja      no_match
    add     cx, dx
    shr     cx, 1       ;(cx+dx)/2
    mov     si, cx
    shl     si,1        ; SI*2→SI
compare:
    add     si, bx      [mid]=[bx+si]
    cmp     ax, es:[si]
    je      exit
    ja      higher
lower:
    dec     cx
    mov     high_idx, cx
    jmp     mid
higher:
    inc     cx
    mov     low_idx, cx
    jmp     mid
no_match:
    stc         ;CF置1
exit:
    pop ds
    ret
b_search  endp
cseg        ends
    end
```



找到：CF=0，SI=相对位置
未找到：CF=1

- 为了提高程序效率
  - 将最常用的数据尽量放在寄存器中
  - 尽量合理分配使用寄存器