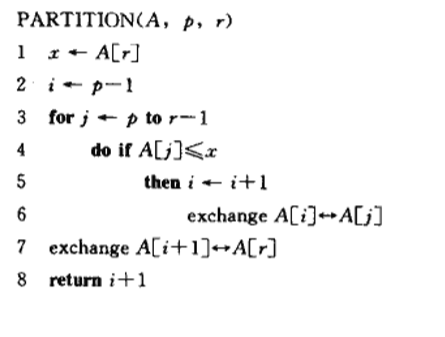
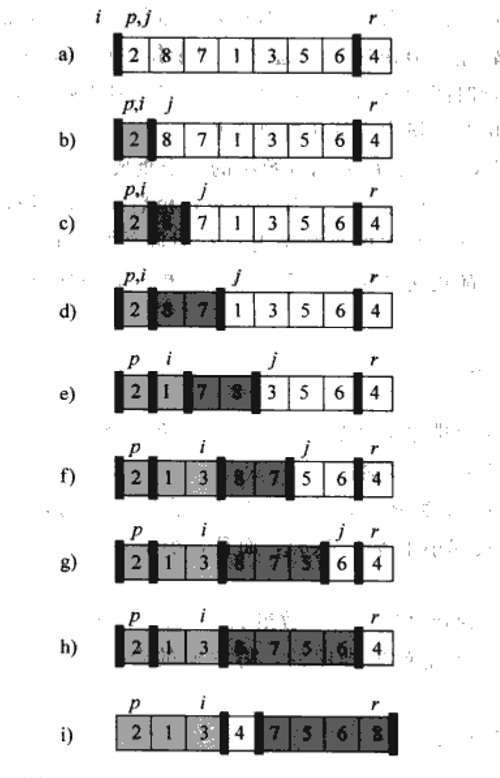
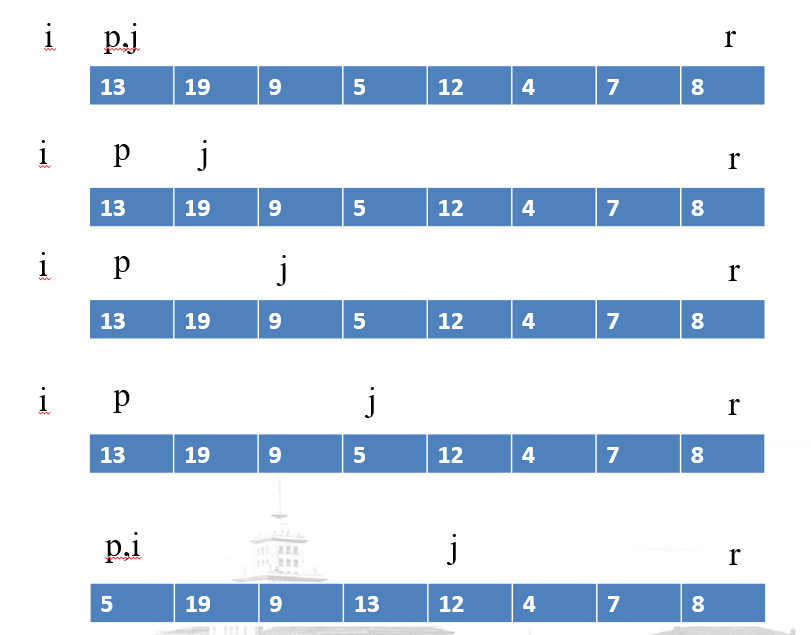
1.（快速排序）以下是快速排序中的一种PARTITION方法的伪代码及过程：

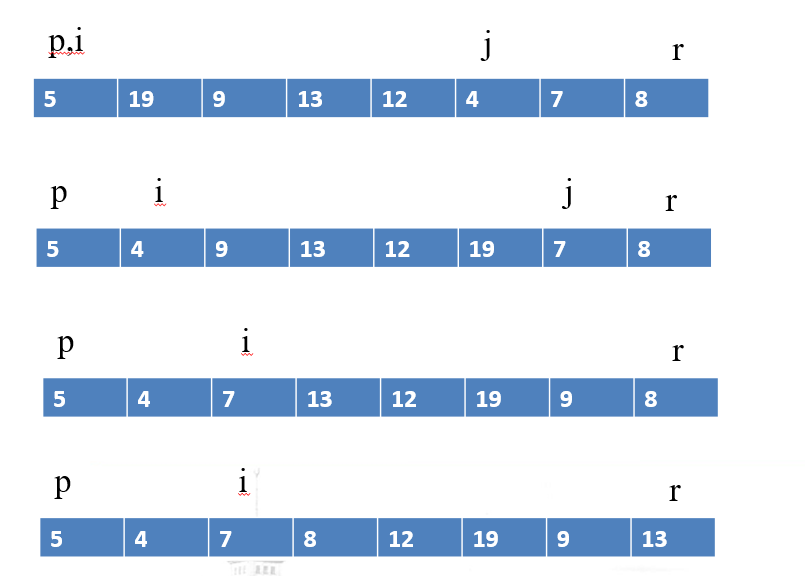


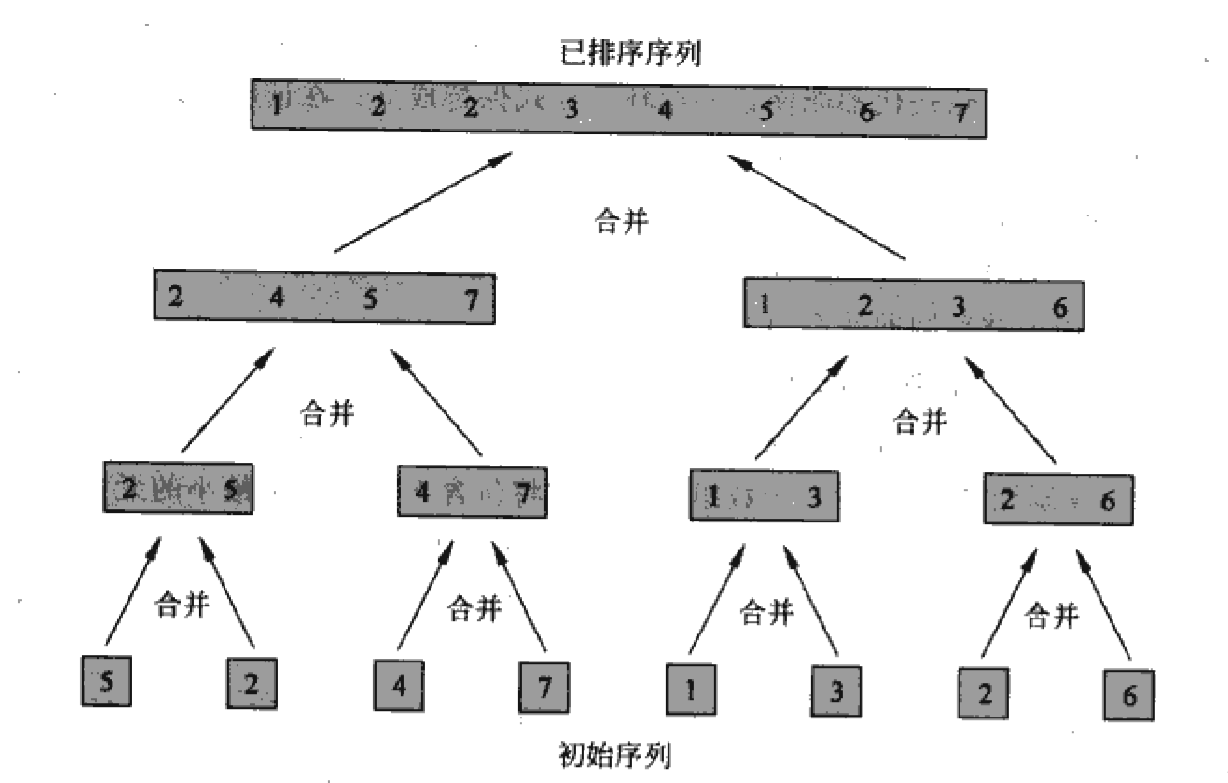


仿照上图说明PARTITION过程作用于数组A=<13,19,9,5,12,4,7,8>的过程。

答案：

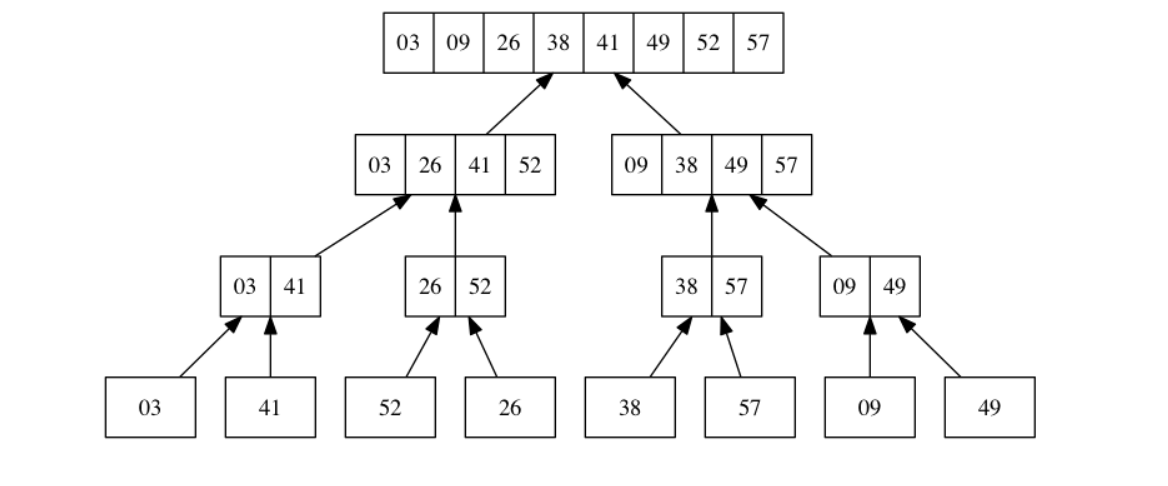






以上图为模型，说明合并排序在输入数组A = <3,41,52,26,38,57,9,49>上的执行过程。

答案：



3.假设A和B是长度为n排好序的数组，且数组中每个数都是不同的。

1. 设计一个算法，在时间里找出这2n个数的中位数，其中2n个数的中位数为从小到大排序的第n个数。
2. 证明你的算法复杂度为

答案：

（1）

**1 lefBegin←0, lefEnd←n-1**

**2 rigBegin←0, rigEnd←n-1**

**3 while(lefBegin < lefEnd && rigBegin <rigEnd){**

**4 if(lefBegin == lefEnd-1 && rigBegin == rigEnd-1) break;**

**5 lefMid ← (lefBegin + lefEnd)/2**

**6 rigMid ← (rigBegin + rigEnd)/2**

**7 if(A[lefMid] == B[rigMid]) return A[lefMid]**

**8 elseif(A[lefMid] < B[rigMid])**

**9 lefBegin←lefMid, rigEnd←rigMid;**

**10 elseif(A[lefMid] > B[rigMid])**

**11 lefEnd←lefMid, rigBegin←rigMid;**

**12 if(lefBegin == lefEnd && rigBegin == rigEnd)**

**13 return min(A[lefBegin], B[rigBegin])**

**14 elseif(lefBegin == lefEnd-1 && rigBegin == rigEnd-1)**

**15 { if(A[lefBegin] < B[rigBegin])**

**16 return min(A[lefEnd],B[rigBegin]);**

**17 else**

**18 return min(A[lefBegin],B[rigEnd]);**

**19 }**

1. **T(n)=T(n/2)+1,T(1)=1**

**根据主定理：a=1,b=2，f(n)=1=。所以T(n)=**

**也就是：T(n)=O(logn)。**

假币问题n枚硬币，其中有一枚是假币，己知假币的重量较轻。现只有一个天平，要求用尽量少的比较次数找出这枚假币。我们用f(A,first,last)函数来完成上述功能。请写出该函数的伪代码 (其中A表示硬币数组[1..n]，first，last为当前考虑的硬币数组中的第一个和最后一个下标，函数返回值为假币的下标)。

答案：

**问题分析：**

将 n 枚硬币分成相等的两部分:

(1)当 n 为偶数时，将前后两部分，即 1...n/2 和 n/2+1... n，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币:

(2)当 n 为奇数时，将前后两部分，即 1..(n -1)/2 和(n+1)/2+1...n，放在天平的两端，较轻的一端里有假币，继续在较轻的这部分硬币中用同样的方法找出假币： 若两端重量相等，则中间的硬币，即第(n+1)/2 枚硬币是假币。

**伪代码：**

f(A，first，last)

firstSum ←0， lastSum ← 0;  
 if first==last-1 then

if A[first] < A[last] then  
 return first;  
return last;

if (last - first + 1) % 2 =0 then

for i ←first to (first+last)/2 do

firstSum+= A[i];  
 for i←first + (last-first) / 2 + 1 to last do

lastSum += A[i];  
 if firstSum<lastSum then

return f(A,first,first+(last-first)/2);  
else

return f(A,first+(last-first)/2+1,last);

else

for i←first to first+(last-first)/2-1 do

firstSum+=A[i];  
 for i←first+(last-first)/2+1 to last do  
 lastSum+=A[i];

if firstSum<lastSum then

return f(coins,first,first+(last-first)/2-1);  
 else if firstSum>lastSum then

return f(coins,first+(last-first)/2+1,last);  
 else

return (first+last)/2;

**假设给定一个不同整数组成的已经排好序的数组A[1,...,n],我们需要在该数组中查找是否存在索引i，使得A[i]=i。**

**（1）尝试用描述分治算法来解决该问题。要求写出伪代码。**

**（2）使用主定理估计第（1）小题中你所描述算法的复杂度。（注意：给出的算法应当保证在的运行时间内）。**

答案：

