第二章 算法基础

2.1 插入排序

1. **以图2-2为模型,说明INSERTION-SORT在数组A=31,41,59,26,41,58上的执行过程**
   答:

序号:	1	2	3	4	5	6
	31	41	59	26	41	58
	31	41	59	26	41	58
	26	31	41	59	41	58
	26	31	41	41	59	58
	26	31	41	41	58	59

2. **重新过程INSERTION-SORT,使之按非升序(而不是非降序)排序**

```
INSERTION-SORT(A)
for j ← 2 to length[A]
    do key ← A[j]
       i = j − 1
       while i > 0 and A[i] < key //将判断大于改为小于即可.
           do A[i + 1] = A[i]
              i = i − 1
       A[i+1] = key
```

3. **考虑以下查找问题:**
   输入: n个数的一个序列A=¡a1,a2,...an¿和一个值v
   输出: 下标i使得v=A[i]或者当v不在A中出现时,v为特殊值NIL.
   写出线性查找的伪代码,它扫描整个序列来查找v,使用一个循环不变式来证明你的算法是正确的.确保你的循环不变试满足三条必要的性质.
   答:

```
FIND-V(A,v)
i = 0
while i < A.length and A[i] ≠ v
    i += 1
    if i < A.length
        OUT i
    else
        OUT NIL
```

正确性:

初始化:

$i = 0$, while 循环迭代前不变式成立,

保持:

i 从0 递增到A.length. 如果A[i] = v,则循环体结束.

终止:

退出循环后,判断i是否在A.length里,如果在,则输出i,否则i为NIL.

4. **考虑把两个n位二进制整数加起来的问题,这两个整数分别存储在两个n元数组A 和B中,这两个整数的和应按二进制形式存储在一个(n+1)元数组C中,请给出该问题的形式化描述,并写出伪代码.**

答:

因为C为n+1位,故可以在C[n+1]存放A+B需要进位的数. 将C初始化为0,

当A+B+C大于2时,说明需要进位,故C[n+1]被置为1,而当前位为(A+B+C)%2;

当A+B+C小于2时,说明不需要进位,当前位等于A+B

```
SUM(A,B,C,N)
C[0 → n + 1]=0
for i = 0 to n
    if A[i] + B[i] + C[i] ≤ 2  //需要进位
      C[i] = (A[i] + B[i] + C[i])%2  //当前位取2的模
      C[i+1] = 1; //实现进位
    else
      C[i] = A[i] + B[i]; // 0 或1
```

2.2 分析算法

1. **用Θ记号表示函数** $n^3 \div 1000 - 100n^2 - 100n + 3$

答:

$\Theta(n^3)$

2. **考虑排序存储在数组A中的n个数:首先找出A中的最小元素并首先找出A中的最小元素并将其与A[1]中的元素进行交换.接着,找出A中的次最小元素并将其与A[2]中的元素进行交换.对A中前n-1个元素按该方式继续.该算法称为选择算法,写出其伪代码.该算法维持的循环不变式是什么?为什么它只需要对前n-1个元素,而不是对所有n个元素运行?用Θ记号给出选择排序的最好情况与最坏情况运行时间.**

答:

```
FUNC(A)
for i = 1 to A.length
   min = i
   for j = i + 1 to A.length //寻找min
       if A[j] ≤ A[min]
          min = j //更新min
   A[i]=A[min]
```

3. **再次考虑线性查找问题(参见练习2.1-3.假定要查找的元素等可能地为数组中的任意元素,平均需要检查输入序列的多少元素?最坏情况又如何?用Θ记号给出线性查找的平均情况和最坏情况运行时间.证明你的答案.**

   答:

   1.平均需要检查输入序列的n/2个元素

   2.最坏需要查找n个元素

   3.平均情况和最坏情况运行时间为Θ(n)

4. **应如何修改任何一个算法,才能使之具有良好的最好情况运行时间?**

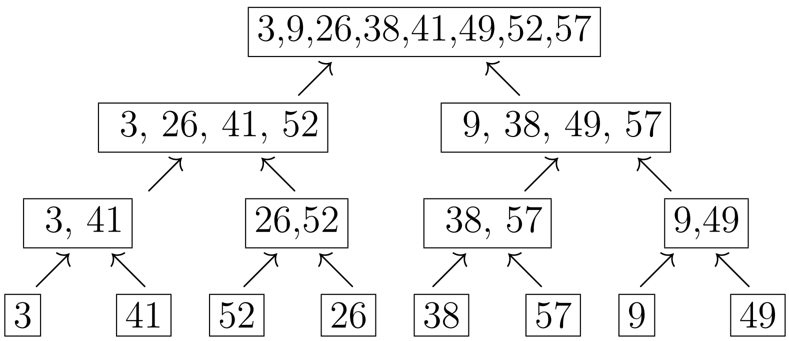
   答:

   针对最大概率的输入定制优化一个算法，可获得最佳运行时间。

2.3 设计算法

1. **使用图2-4作为模型,说明归并排序在数组A={3,41,52,26,38,57,9,49}上的操作;**

   答:



2. **重写过程MERGE,使之不使用哨兵,而是一旦数组L或R的有元素均被复制回A就立刻停止,然后把另一个数组的剩余部分复制回A**

答:

```
MERGE(A,p,q,r)
n1 = q - p + 1
n2 = r - q
let L[1 → n1+1] and R[1 → n2+1] be new arrays
for i = 1 to n1
    L[i] = A[p+i-1]
for j = 1 to n2
    R[j] = A[q+j]
i = 1;  j = 1; k = p
while i ≤ n1 and j ≤ n2
     if L[i] ≤ R[j]
         A[k] = L[i]
         i += 1
     else
         A[j] = R[j]
         j += 1
     k += 1
if i ≤ n1
    while i ≤ n1
         A[k] = L[i]
         k += 1
         i += 1
else
    while j ¡ n2
         A[k] = R[j]
         k += 1
         i += 1
```

3. **使用数学归纳法证明：当n刚好是2的幂时,以下递归的解是T(n)=nlgn.**

$$\mathrm{T(n)}=\begin{cases} 2 & \text{若 n=2} \\ 2\mathrm{T(n/2)+n} & \text{若 } \mathrm{n} = 2^k, k > 1 \end{cases}$$

答:

n = 2时，

T(n) = 2 成立。

假定当n=2^k，$k > 1$时成立，有：

T(n) = $2^k lg(2^k) = (2^k) * k$