习题参考答案

1.1

最优解决例子：百米比赛中金牌获得者的判断方法，只能是跑最短时间的那个选手。

“ 近似 ” 最优解例子：通常数学解题中用 π 的近似值 3.14 来代替 π 。

1.2

在该算法中，当整数集 S 确定后，它的所有子集也就确定了，所以是否含有和为 m 的子集也就可以求出来，该算法是确定的，可行的。另外集合 S 的元素个数 n 是有穷的，而它的子集的总个数为 2^n ，同样也是有穷的。所以算法也是有穷的。综上可得该算法满足算法的特点。

1.3

15	12	1	3	7	19	12	15
12	15	1	3	7	19	12	15
1	12	15	3	7	19	12	15
1	3	12	15	7	19	12	15
1	3	7	12	15	19	12	15
1	3	7	12	15	19	12	15
1	3	7	12	12	15	19	15
1	3	7	12	12	15	15	19

1.4

```
InsertSort(A)
for  j←2 to n do
        key ← A[j]
        i← j-1
        while i > 0 and A[i] < key do      //修改地方
              A[i + 1]← A[i]
              i ← i -1
        A[i + 1] ← key
return A
```

1.5

FindMax (A)	cost	times
max← A[1]	c_1	1
for j←2 **to** n **do**	c_2	n
if $A[j]$>max **then**	c_3	n-1

4	max←*A*[*j*]	c_4	$\sum_{2}^{n} t_i$

$$T(n) = c_1 + c_2 * n + c_3 * (n-1) + c_4 * \sum_{2}^{n} t_i$$

当数组 *A* 的第一个元素为最大时该算法达到最佳，此时$t_i = 0$：

$$T(n) = c_1 + c_2 * n + c_3 * (n-1) = (c_2 + c_3) * n + c_1 - c_3 = \Theta(n)$$

当数组 *A* 是递增时该算法达到最坏，此时$t_i = 1$：

$$T(n) = c_1 + c_2 * n + c_3 * (n-1) + c_4 * (n-1) = (c_2 + c_3 + c_4) * n + c_1 - c_3 - c_4 = \Theta(n)$$

1.6

循环不变量：在 **for** 循环第 *j* 个迭代执行前，max 为当前数组 *A*[1..*j*-1]中的最大值

初始步：在第一轮迭代前，*j*=1，当前数组中只有一个元素 max =*A*[1]，此时 max 确是最大的，所以在初始化中不变量是正确的。

归纳步： 在 **for** 循环第 *j*=*k* 个迭代执行前，max 是 *A*[1,...,*k*-1] 中的最大值，此后执行第 *j*=*k* 个迭代，将 max 与 *A*[*k*] 进行比较并将较大的值赋给 max，所以 max 仍然是 *A*[1,..., *k*]中的最大值。

终止步：当 *j*=*n*+1 时，for 循环终止，此时 max 是 A[1,...,*n*] 中的最大元素，即是全数组中的最大值，所以算法是正确的。

1.7

```
MaxIndex(A)
max←A[1]
j←1
for i←2 to n do
    if max≤A[i] then
        max←A[i]
        j←i
return j
```

1.8

```
Exp(a, n)
i←1
pow←1
while i≤n do
    pow←pow×a
    i←i+1
return pow
```

循环不变量：当执行第 *i* 次循环前，pow 的值为 a^{i-1}

初始步：当 i=1，即在第一轮循环前，pow 为 1，若 n=0，则不循环，返回 1，若 n=1，则循环一次，pow=a，所以不变量正确

归纳步：当执行第 i=k 次循环前，假设 pow 的值为 a^{k-1}，当执行第 i=k 次循环后，则由算法可知，pow=pow*a=a^{k-1}*a=a^k，所以不变量仍正确

终止步：当 i=n+1 时，循环退出，此时共执行了 n 次循环，所以结果为 a^n，算法是正确的

1.9

Searchx(A, x)		cost	times
1	i←1	c_1	1
2	**while** x≠A[i] and i≤n **do**	c_2	t
3	i←i+1	c_3	t-1

//其中 t≤n+1

$$T(n) = c_1 + c_2 t + c_3(t-1)$$

当第一个元素就是指定数 x 时该算法达到最好情形，此时 t=1：

$$T(n) = c_1 + c_2 = \Theta(1)$$

当该数组中不含指定数 x 时该算法达到最坏情形，此时 t=n：

$$T(n) = c_1 + c_2(n+1) + c_3 n = \Theta(n)$$

正确性证明：

循环不变量：当执行第 i 次循环前，指定数 x 不在当前数组 A[1..i-1]中

初始步：当 i=1，即在第一轮循环前，当前数组为空，不包含指定数，所以循环不变量成立。

归纳步：当执行第 i=k 次循环前，指定数 x 不在当前数组 A[1..k-1]中，当执行第 i=k 次循环后，此时表明 x≠A[i]且 i≤n，否则，找找到了指定数。因此，在执行第 i=k+1 次循环前，指定数 x 不在当前数组 A[1..k]中，所以循环不变量成立。

终止步：当 i=n+1 时，循环退出，此时，表明指定数 x 不在当前数组 A[1..n]中。否则，在前面任一次循环退出，均找到了指定数，所以整个算法是正确的。

1.10

要使运行时间 100*n^2 比 2^n 快，则需要：

$$100*n^2 < 2^n$$

通过计算得 $n \geq 15$

所以得 n 最小值为 15

1.11

要使插入排序优于合并排序，则：

$$8n^2 < 64n\lg n$$

$$\Rightarrow n/8 < \lg n$$

$$\Rightarrow 2^{n/8} < n$$

$$\Rightarrow 2 \le n \le 43$$ // 这两个值是通过计算得来的

实验题

1.11 完成 XOJ 如下题目：1000，1001，1002，1003，1004。

1000 A+B 1001 宅男健身计划 1002 C=?+? 1003 Sort 1004 Sort Ver.2

注：1000 1001 1003 必练习，1002 和 1004 可不做