李青航 SA22225226

2.1-2

Algorithm 1 No Increasing INSERTION-SORT(A)

```
for j = 2 to A.length do
    key = A[j]
    i = j − 1
    while i > 0 and A[i] < key do
        A[i + 1] = A[i]
        i = i − 1
    end while
    A[i + 1] = key
end for
```

2.1-3

初始化：$j = 1, A.length = 1$时，要么$A[1]$ =value，返回对应位置$i = 1$,要么没找到返回NIL，显然正确。

保持：不断迭代，当碰到$A[j] = v$时（第5行）返回对应位置i。

终止：当迭代完所有j后，没有找到value，返回NIL。

Algorithm 2 Linear-Search(A, v)

```
1: i = NIL
2: for j = 1to A.length do
3:     if A[j] = v then
4:         i = j
5:         return i
6:     end if
7: end for
8: return i
```

2.2-1

$n^3/1000 - 100n^2 - 100n + 3$表示为$\Theta(n^3)$

2.2-2

Algorithm 3 Selection Sort(A)

```
for i = 1 to n − 1 do
    min = i
    for j = i + 1 to n do
        if A[j] < A[min] then
            min = j
        end if
    end for
    Swap(A[min], A[i])
end for
return i
```

初始化：当$i = 1$,数组A中也只有一个元素时，显然已经排序，正确。
保持：当迭代到寻找第i小的元素时，$A[1...i-1]$已经排序好，将第i小的与$A[i]$交换，此时，$A[1...i]$都排序好了。
终止：当进行到$n-1$次时，前$A[1...n-1]$已经非递减有序，第n个就是最大，整体全部排好，终止，正确。

只用进行n-1次，因为最后一个（第 n个）最大就应该在最后，全部已经排序好。

最好情况$\Theta(n^2)$，最坏情况$\Theta(n^2)$

3.1-1
存在正常量$c_1 = 0.5, c_2 = 1, n_0$,当$n \geq n_0$时，$0.5(f(n)+g(n)) \leq max(f(n), g(n)) \leq f(n) + g(n)$，符合$\Theta(f(n) + g(n))$定义

3.1-2
根据多项式定理$(x_1 + x_2 + \cdots + x_m)^b = \sum_{k_1+k_2+\cdots+k_m=n} \frac{n!}{k_1! \cdot k_2! \cdot \cdots \cdot k_m!} \prod_{t=1}^{m} {x_t}^{k_t}$，
二项式的b次方展开公式,$(n + a)^b = n^b + ...$其中省略的...部分是n^b的低阶，所以根据Θ记号定义，多项式可以省略低阶，所以 $(n + a)^b = \Theta(n^b)$