Chapter 2 Homework

Deadline: 2021/09/24 10:10 a.m.

1.

Find the time complexity of the following code.

```
for(i=1; i<=N; i++)
    for(j=i; j<=N; j+=i)
        // some operations in constant time
```

2.

Someone designs his/her insertion sort algorithm as follows. Can the code work in any situation? If it can, briefly explain its correctness. If it cannot, give a counterexample.

```
for j ← 2 to A.length
    key ← A[j]
    for i ← j-1 to 1:
        if A[i]>key:
            A[i+1] ← A[i]
        else:
            A[i+1] ← key
            break
```

3.

Given an integer array *A*, *p = 1*, *r = A.length*, please write pseudocode to calculate the number of reverse pairs in the array.

A reverse pair is a pair *(i, j)* where *1 <= i < j <= A.length* and *A[i] > A[j]*.

For example, if the array is [1, 5, 2, 4, 3], then the answer is four because there are four reverse pairs (5, 2), (5, 4), (5, 3), (4, 3).

The preferred time complexity of your algorithm is $O(N \log N)$, while $O(N^2)$ algorithm can also get partial points.