Course: Algorithm
Prof. Prem Nair
Student: Binh Van Tran
ID: 986648
Homework: Lab 2

1. **Question 1 – Comparing Algorithm**
   Write pseudo code for the following algorithms:
   - Algorithm 1: Create a new array consisting of even numbers only. Then use nested loops to solve the problem using the newly created array of even numbers only

```
findLargestDistance(A, n)
    Input array A of n integers
    Output largest distance of two even elements
    maxDistance ← INT_MINVALUE
    for i ← 0 to n − 1 do
        if A[i] modulo 2 == 0 then
            evens.append(A[i])
    evenLength ← evens.LENGTH
    for i ← 0 to evenLength− 2 then
        for j ← i + 1 to evenLength − 1 then
            if maxDistance < Absolute(evens[i] − evens[j]) then
                maxDistance ← Absolute(evens[i] − evens[j])
    return maxDistance
```

   - Algorithm 2: Use a nested loop to solve the problem without creating an extra array

```
findLargestDistance(A, n)
    Input array A of n integers
    Output largest distance of two even elements
    maxDistance ← INT_MINVALUE
    for i ← 0 to n − 2 then
        if A[i]%2 != 0 then
            continue
        for j ← i + 1 to n − 1 then
            if A[j]%2 != 0 then
                continue
            if maxDistance < Absolute(A[i] − A[j]) then
                maxDistance ← Absolute(A[i] − A[j])
    return maxDistance
```

- Algorithm 3: Use one loop. Find max and min of even integers. Compute max – min

```
findLargestDistance(A, n)
    Input array A of n integers
    Output largest distance of two even elements
    max ← INT_MAXVALUE
    min ← INT_MINVALUE
    for i ← 0 to n – 1 then
        if A[i]%2 == 0 then
            if max < A[i] then
                max ← A[i]
            if min > A[i] then
                min ← A[i]
    return max − min
```

As we can easily determine the worst-case time complexity:
Algorithm 1 & 2 is $O(n^2)$
Algorithm 3 is $O(n)$
This is consistent because in W1D1 Algorithm 1 & 2 time to finish is getting bigger when the size of input array is getting bigger. While Algorithm 3 is not.
W1D1: We can know which algorithm is better by practical experiment
W1D2: We can know which algorithm is better based on its time complexity

2. **Question 2 – Complete the table:**

10, 1	$O(1)$
$\log n$, $\log^2$n, ln$n$, $n^{1/2}$logn, $n^{1/3}$logn	$O(\log n)$
$n^{1/k}$ (k>3)	$O(n^{1/k})$
$n^{1/2}$, $n^{1/3}$	$O(n)$
nlog$n$, logn^n	$O(n\log n)$
n^3, n^2, n^k	$O(n^k)$
2^n, 3^n	$O(2^n)$
n!	$O(n!)$
n^n	$O(n^n)$