**Luyện tập tại lớp**

**Exercise 1.**

Phân tích:

Input: 1 mảng n phần tử a (a0,a1,...,an) tương đương với n bậc, giá trị x0

Output: giá trị của đa thức bậc n

a)

Mã giả:

**valuePolynomial**(As, x):int≡

n = len(As)

sum = 0

**for** ( i= 0 .. n-1)

xPow = 1

**for** (j= 0 .. i-1)

xPow += x

sum += As[i] \* xPow

return sum

**end.**

=> Độ phức tạp thời gian O(n2)

b)

Mã giả:

**valuePolynomial**(As, x):int≡

n = len(As)

sum = 0

xPow = 1

**for** (i = 0 .. n-1)

sum += As[i] \* xPow

xPow \*= x

return sum

**end**.

=> Độ phức tạp thời gian O(n)

c) Không tồn tại thuật toán với độ phức tạp thời gian nhỏ hơn O(n) cho bài toán này vì để tính được đa thức ta phải duyệt tối thiểu qua n phần tử ( đa thức gồm n phần tử).

**Exercise 2.**

Mã giả:

**selectionSort(array):**array≡

n = len(array)

**for** i in range(n-1)

minIndex = i

**for** j in range(i+1, n)

**if** array[j] < array[minIndex]

minIndex = j

**endif**

array[i], array[minIndex] = array[minIndex], array[i]

print(array)

return array

**end.**

Đầu vào và array qua từng bước.

array = H, O, M, E, W, R, K

Bước 1 [E, O, M, H, W, R, K]

Bước 2 [E, H, M, O, W, R, K]

Bước 3 [E, H, K, O, W, R, M]

Bước 4 [E, H, K, M, W, R, O]

Bước 5 [E, H, K, M, O, R, W]

Bước 6 [E, H, K, M, O, R, W]

**Exercise 3.**

**Mã giả:**

**bubbleSort(array):**array≡

n = len(array)

**for** i in range(n)

**for** j in range(i+1, n)

**if** array[i] > array[j]

temp = array[i]

array[i] = array[j]

array[j] = temp

**endif**

return array

**end**.

Đầu vào và array qua từng bước.

array = H, O, M, E, W, R, K

Bước 1 [E, O, M, H, W, R, K]

Bước 2 [E, M, O, H, W, R, K]

Bước 3 [E, H, O, M, W, R, K]

Bước 4 [E, H, M, O, W, R, K]

Bước 5 [E, H, K, O, W, R, M]

Bước 6 [E, H, K, M, W, R, O]

Bước 7 [E, H, K, M, R, W, O]

Bước 8 [E, H, K, M, O, W, R]

Bước 9 [E, H, K, M, O, R, W]

**Exercise 4.**

Input: số n cố định

Output: giá trị p sao cho C(n) min, và p sao cho C(n) max.

**Mã giả:**

**valuePForMinMax(n)** ≡

max = ‘inf’

min = ‘-inf’

**for** p in range(0, 101)

p = p / 100.0

cN= p \* (n + 1) / 2 + n \* (1 – p)

**if** cN > max

max = cN

pForMax = p

**endif**

**if** cN < min

min = cN

pForMin = p

**endif**

return pForMin, pForMax

**end.**

Do hệ số n – (n -1) p/ 2 => hàm này nghịch biến nên mã giả bên trên là đúng