

Câu 1:

Độ phức tạp : $O(n*m)$

Câu 2:

void impBubbleSort(int a[], int n)

```
{  
    int temp,count=0;  
    for (int i = 1; i <= n - 1; i++)  
    {  
        for (int j = n - 1; j > i; j--)  
        {  
            if (a[j - 1] > a[j])  
            {  
                temp = a[j - 1];  
                a[j - 1] = a[j];  
                a[j] = temp;  
                count++;  
            }  
        }  
        if (count == 0)  
        {  
            break;  
        }  
    }  
}
```

```
}
```

Câu 3:

Giải thuật có độ phức tạp  $O(n+k)$  :Counting Sort

```
int Max(array, n)
```

```
{
```

```
    int max = giá trị đầu của mảng
```

```
    for (int i = 2; nếu i còn bé hơn n; i tăng lên 1 giá trị)
```

```
    {
```

```
        if (giá trị tại index i > max)
```

```
            max = array[i]
```

```
    }
```

```
    return max
```

```
}
```

```
void CountingSort(int array[], int n)
```

```
{
```

```
    int output[n + 1];
```

```
    int max = Max(array, n);
```

```
    int count[max + 1];
```

```
    for (int i = 0; i còn bé hơn hay bằng max; i tăng lên 1 giá trị)
```

```
        count[i] = 0;
```

```
    for (int i = 1; i còn bé hơn hay bằng n; i tăng giá trị lên 1)
```

```
count[array[i]]++;
```

```
for (int i = 1; i còn bé hơn hay bằng max; i tăng lên 1 giá trị)
```

```
count[i] += count[i - 1];
```

```
for (int i = n; i còn lớn hơn n; i giảm 1 giá trị)
```

```
{
```

```
    output[count[array[i]]] = array[i];
```

```
    count[array[i]] -= 1;
```

```
}
```

```
for (int i = 1; i còn bé hơn hay bằng n; i tăng lên 1 giá trị)
```

```
{
```

```
    array[i] = output[i];
```

```
}
```

```
}
```



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

## THẺ SINH VIÊN



**BIỆN VIẾT PHẨM**

Ngày sinh : 27/02/2002

MSSV: 20120155

Khoa : Công nghệ Thông tin

Bậc : Đại học



Hạn thẻ: 2020 - 2024

[www.hcmus.edu.vn](http://www.hcmus.edu.vn)