

Bài 1

Cho mã giả các thuật toán sau, hãy viết chương trình python cho từng thuật toán

```
def cocktail_sort(array: a):
    begin, end = 0, len(a) - 1
    while (begin < end):
        # sắp xếp nổi bọt theo chiều tăng dần
        for i = begin to (end - 1):
            if a[i] > a[i + 1]:
                swap(a[i], a[i + 1])
        # giảm end do phần tử lớn nhất đã về đúng vị trí
        end = end - 1
        # sắp xếp nổi bọt theo chiều giảm
    for j = end downto (begin + 1):
        if a[j] < a[j - 1]:
            swap(a[j], a[j - 1])
    # tăng begin do phần tử nhỏ nhất đã về đúng vị trí
    begin = begin + 1
```

```
def shell_sort(array: a, array: gaps):
    for g in gaps:
        for i = g to len(a) - 1:
            temp = a[i]
            k = i - g
            while (k >= 0 and a[k] > temp):
                a[k + g] = a[k]
                k = k - g
            a[k + g] = temp
```

```

# hàm phân hoạch mảng từ vị trí low đến vị trí high
# phần tử chốt được chọn là giá trị a[high]
# trả về vị trí phân hoạch
def partition(array: a, int: low, int: high):
    # lấy giá trị chốt
    pivot = a[high]
    i = low
    for j = low to (high - 1):
        # đổi chỗ các phần tử nhỏ hơn giá trị chốt về đầu
        # mảng
        if a[j] < pivot:
            swap(a[i], a[j])
            i = i + 1
    # đổi chỗ a[high] về vị trí phân hoạch
    swap(a[i], a[high])
    return i
# hàm sắp xếp nhanh
# để sắp xếp mảng a, gọi quicksort(a, 0, len(a) - 1)
def quicksort(array: a, int: low, int: high):
    if low < high:
        # phân hoạch mảng a
        p = partition(a, low, high)
        # sau phân hoạch, a[p] đã ở đúng vị trí
        # sắp xếp nhanh phân hoạch trái và phải
        quicksort(a, low, p - 1)
        quicksort(a, p + 1, high)

```

Bài 2

Trong python, để đo thời gian thực hiện một khối lệnh, ta có thể dùng đến cấu trúc sau:

```

%%timeit -o
<khối_lệnh_cần_đo_thời_gian>
res = _
print(res.average * 1e9)

```

Dùng khối lệnh trên và đo thời gian thực hiện các thuật toán ở Bài 1 với dữ liệu được cho trong 3 file 'random.txt', 'nearly__sort.txt' và 'reversed.txt'.