

ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

NHẬP MÔN TÍNH ĐỒ PHỨC TẠP THUẬT TOÁN

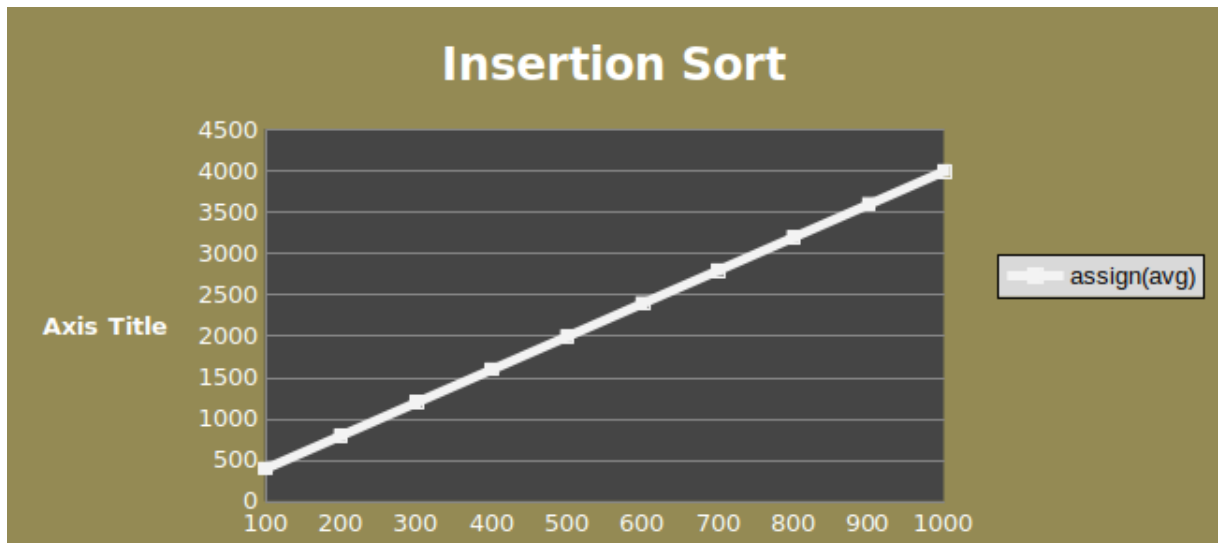
BÀI TẬP TUẦN 3

Họ và tên	MSSV
Lê Nguyễn Nhật Trường	1712195



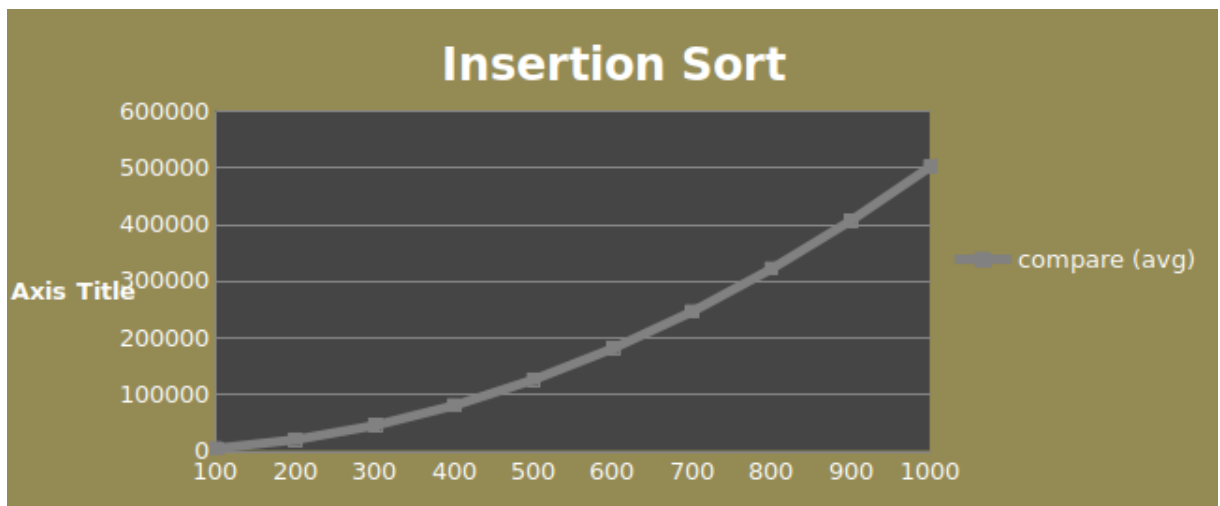
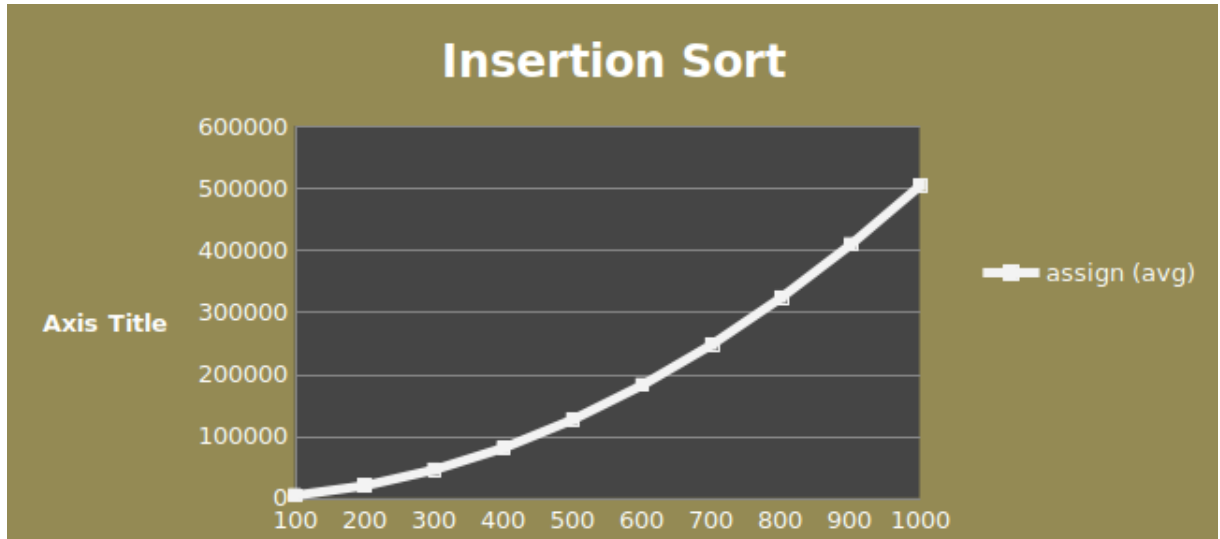
1 Trường hợp tốt nhất

Theo code thuật toán insertion sort đạt trường hợp tốt khi mảng đã sắp xếp giảm dần. Do việc sắp xếp giảm dần làm giảm số thao tác assign và compare rất nhiều. Vì ở đây chúng ta đang sắp xếp giảm dần.



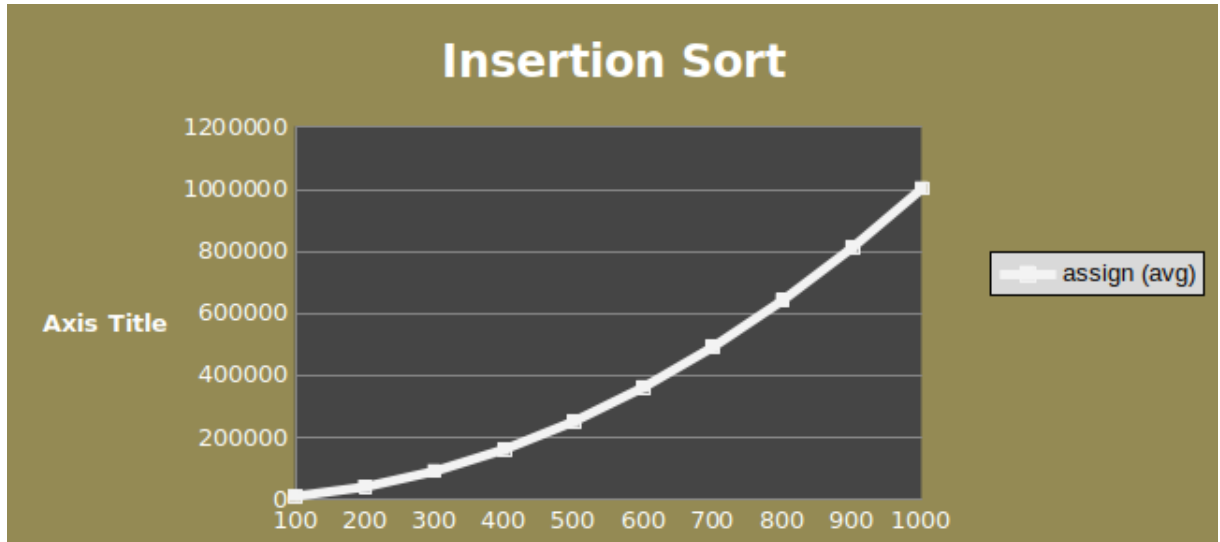
2 Trường hợp trung bình

Theo code thuật toán insertion sort đạt trường hợp trung bình khi mảng đã ngẫu nhiên.



3 Trường hợp xấu nhất

Theo code thuật toán insertion sort đạt trường hợp trung bình khi mảng đã sắp xếp rồi.



4 So sánh với lý thuyết

```
int i, j, key, numLength;

numLength = arr.size(); countAssign++;

for(j = 1, countAssign++; ++countCompare && j < numLength;
    j = j + 1, countAssign++)
{
    key = arr[j];
    countAssign++;

    for(i = j - 1, countAssign++; (++countCompare && i >= 0)
        && (++countCompare && arr[i] < key); i = i - 1, countAssign++)
    {
        arr[i+1] = arr[i]; countAssign++;
    }

    arr[i+1] = key; countAssign++;
}
```

Gọi $A(n)$ là số phép assign của thuật toán insertion sort.
 $C(n)$ là số phép compare của thuật toán insertion sort.

4.1 Trường hợp tốt nhất

$$A(n) = 1 + (4 \times (n - 1) + 1) = 4 \times n - 2$$
$$C(n) = (3 \times (n - 1) + 1) = 3 \times n - 2$$

4.2 Trường hợp xấu nhất

$$A(n) = 1 + (2 \times (n - 1) + 1) + \sum_{j=1}^{n-1} (2 \times j + 2)$$
$$A(n) = 1 + (2 \times (n - 1) + 1) + n \times (n + 1)$$

$$A(n) = n^2 + 3n - 2$$

$$C(n) = n + (n - 1) \times (n + 1)$$

$$C(n) = n^2 + n - 1$$

4.3 Trường hợp trung bình

Trường hợp trung bình cũng tương tự trường hợp xấu nhất.

$$A(n) = 1 + (2 \times (n - 1) + 1) + \sum_{j=1}^{n-1} (\times j + 1)$$

$$A(n) = 1 + (2 \times (n - 1) + 1) + \frac{n \times (n + 1)}{2}$$

$$A(n) = \frac{n^2 + 5n}{2}$$

$$C(n) = n + \frac{(n - 1) \times (n + 1)}{2}$$

$$C(n) = \frac{n^2 + 2n - 1}{2}$$

5 Tổng kết

- Sắp xếp chèn là một thuật toán sắp xếp đơn giản, xây dựng một mảng được sắp xếp cuối cùng một mục tại một thời điểm. Nó kém hiệu quả hơn nhiều trong các danh sách lớn so với các thuật toán tiên tiến hơn như quicksort, heapsort hoặc hợp nhất sắp xếp.
- Độ phức tạp tồi tệ nhất: $O(n^2)$
- Độ phức tạp trung bình: $O(n^2)$
- Độ phức tạp tốt nhất: $O(n)$
- Độ phức tạp không gian: 1