Short note: Insertion sort

```
def insertion_sort(a):
    for p in range(1, len(a)):
        tmp = a[p]
        j = p
        while j > 0 and tmp < a[j - 1]:
            a[j] = a[j - 1]
            j -= 1
        a[j] = tmp
```

```
my_list = [34, 8, 64, 51, 32, 21]
insertion_sort(my_list)
```

a

34	8	64	51	32	21
[0]	[1]	[2]	[3]	[4]	[5]

For ของ p ; p = 1 ; 1 < 6 จริง

tmp =

j

while ของ j ; j = ; > 0 && tmp < a[j - 1] = ;

34 8 64 51 32 21

while ของ j ; j = ,; 0 > 0 เท็จ && tmp < a[j - 1];

After p = 1 8 34 64 51 32 21 position move = ?

[0] [1] [2] [3] [4] [5]

For ของ p ; p = ; < 6 จริง

tmp =

j

while ของ j ; j = ; > 0 && tmp < a[j - 1] =;

After p = 2 8 34 64 51 32 21

[0] [1] [2] [3] [4] [5]

For ของ p ; p = ; < 6 จริง

tmp =

j

while j = , ; 2 > 0 && tmp < a[j - 1] =

a[j] = ?

After p = 3 8 34 51 64 32 21

[0] [1] [2] [3] [4] [5]

For ของ p ; p = ? ; ?<6

tmp= ?

After p= 8 34 51 64 32 21

[0] [1] [2] [3] [4] [5]

After p = ? 8 34 51 64 32 21 position move = ?

[0] [1] [2] [3] [4] [5]

P Tmp = ?

j

Tmp= ?

- j = p: กำหนดตัวแปร j ให้มีค่าเท่ากับ p เพื่อใช้เป็นตัวชี้ในการเปรียบเทียบและเลื่อนตำแหน่ง

while j > 0 and tmp < a[j - 1]: นี่คือหัวใจของการแทรก

- j > 0: เงื่อนไขนี้ทำให้แน่ใจว่าเราไม่ได้พยายามเข้าถึง index ที่น้อยกว่า 0 (ซึ่งไม่มีในลิสต์)
- tmp < a[j - 1]: เปรียบเทียบค่าของสมาชิกที่เรากำลังจะแทรก (tmp) กับสมาชิกที่อยู่ก่อนหน้า (a[j - 1])
- **ถ้าทั้งสองเงื่อนไขเป็นจริง:** หมายความว่า tmp มีค่าน้อยกว่าสมาชิกที่อยู่ก่อนหน้า ดังนั้น สมาชิกที่อยู่ก่อนหน้า (a[j - 1]) ต้องถูก **เลื่อนไปทางขวา** หนึ่งตำแหน่ง เพื่อเปิดทางให้ tmp
- a[j] = a[j - 1]: เลื่อนสมาชิก a[j - 1] ไปยังตำแหน่ง a[j] อย่าลืม หมายถึง **นำค่าทางขวา มาใส่ทางซ้าย**
- j -= 1: ลดค่า j ลง 1 เพื่อขยับไปเปรียบเทียบกับสมาชิกตัวถัดไปทางซ้าย
- **เมื่อเงื่อนไขใน while เป็นเท็จ (คือ j เป็น 0 แล้ว หรือ tmp ไม่น้อยกว่า a[j - 1] แล้ว) แสดงว่าเราเจอตำแหน่งที่เหมาะสมแล้ว จึงนำค่า tmp ไปใส่ไว้ที่ตำแหน่ง j ด้วยคำสั่งว่า a[j] = tmp**