

# 삽입 정렬이란?

처리되지 않은 데이터를 하나씩 골라 적절한 위치에 삽입  
선택 정렬에 비해 구현 난도가 높은 편이지만, 일반적으로 더 효율적으로 동작

## 삽입 정렬 동작 예시

7 5 9 0 3 1 6 2 4 8

---

7 자체는 정렬이 되어있다는 가정하에 두 번째 데이터인 5가 어떤 위치로 들어갈지 판단한다.

이렇게 7의 외쪽으로 들어가거나 오른쪽으로 들어가거나 두 경우만 존재한다.

5 7 9 0 3 1 6 2 4 8

---

이런 식으로 변경이 된다. 이것을 반복해서 한다.

## 삽입 정렬 소스 코드

```

array = [7,5,9,0,3,1,6,2,4,8]

for i in range(1,len(array)):
    for j in range(i,0,-1): # 인덱스 i 부터 1 까지 1씩 감소하며 반복하는 문법
        if array[j] < array[j-1] # 한 칸씩 왼쪽으로 이동
            array[j],array[j-1] = array[j-1],array[j]
        else: # 자기보다 작은 데이터를 만나면 그 위치에서 멈춤
            break

print(array)

```

## 삽입 정렬의 시간 복잡도

삽입 정렬의 시간 복잡도는  $O(N^2(\text{제곱}))$ 이며, 선택 정렬과 마찬가지로 반복문이 두 번 중첩되어 사용

삽입 정렬은 현재 리스트의 데이터가 거의 정렬되어 있는 상태라면 매우 빠르게 동작

- 최선의 경우  $O(N)$ 의 시간 복잡도를 가진다.

[www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC](http://www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC)

이 자료는 동빈 나 님의 이코 테 유튜브 영상을 보고 정리한 자료입니다.

		84,550원
		35,400원
	155,000원	

## 'Algorithm' 카테고리의 다른 글

**[Algorithm] 22강 : 삽입 정렬**

[Algorithm] 21강 : 선택 정렬

[Algorithm] 20강 : DFS & BFS 기초 문제 풀이

[Algorithm] 19강 : BFS(너비 우선 탐색) 알고리즘 정의와 예제

[Algorithm] 18강 : DFS(깊이 우선 탐색) 알고리즘 정의와 예제

[Algorithm] 17강 : 재귀함수의 정의와 예제