

BubbleSort

| | |
|------------|--------------------------|
| 🕒 작성일시 | @2022년 8월 3일 오전 9:56 |
| ▼ 강의 번호 | |
| ▼ 유형 | |
| 🔗 자료 | |
| ☰ Property | |
| 📅 Date | |
| ☑ 복습 | <input type="checkbox"/> |
| ☰ 속성 | |

▼ 단순 교환법 Bubble Sort

▼ 설명

- 인접한 데이터를 교환하는 처리를 반복하여 전체를 정렬한다.
- 아주 단순하여 이해하기 쉬운 알고리즘이지만 실행 속도가 느리다.

단순 교환법은 버블 정렬이라고 부르고 말 그대로 버블은 거품을 의미한다.

마치 거품이 수면위로 올라와서 이동하는 것 처럼 보인다고 해서 붙여진 이름이다.

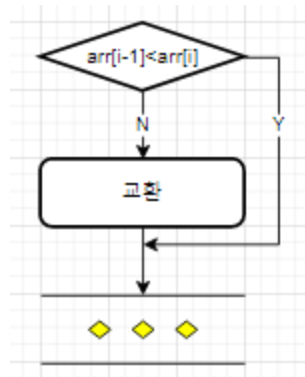
- 왼쪽 끝 칸부터 순서대로 들어갈 공을 확정시켜 나간다.

arr[0]=5, arr[1]=3, arr[2]=4....

- 오른쪽 끝부터 순서대로 인접한 공을 오름차순으로 정렬한다.
 1. 인접한 요소의 데이터를 비교
 2. 오름차순으로 바꾸어 나열하는 처리

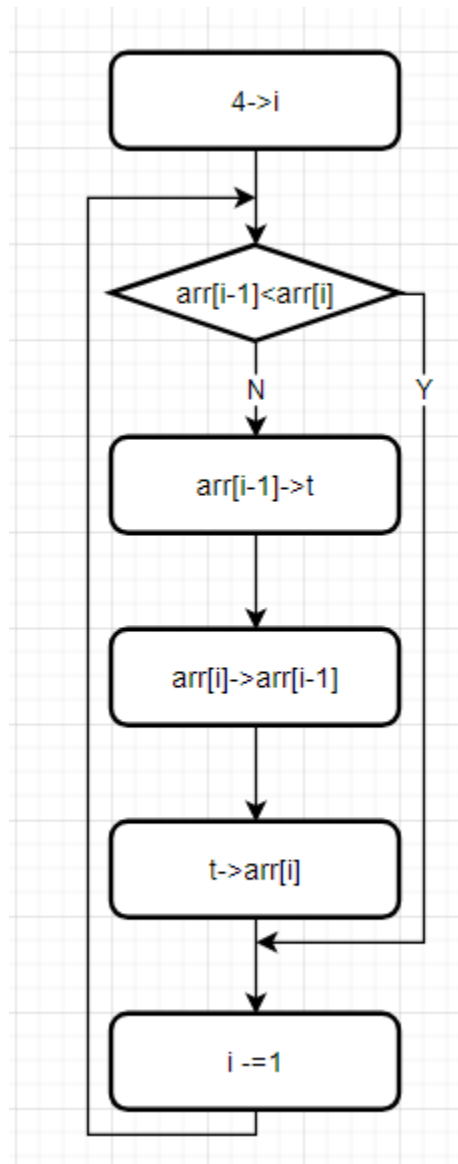
| | 왼쪽요소 | 오른쪽요소 |
|-----|----------|--------|
| 첨자 | i-1 | i |
| 데이터 | arr[i-1] | arr[i] |

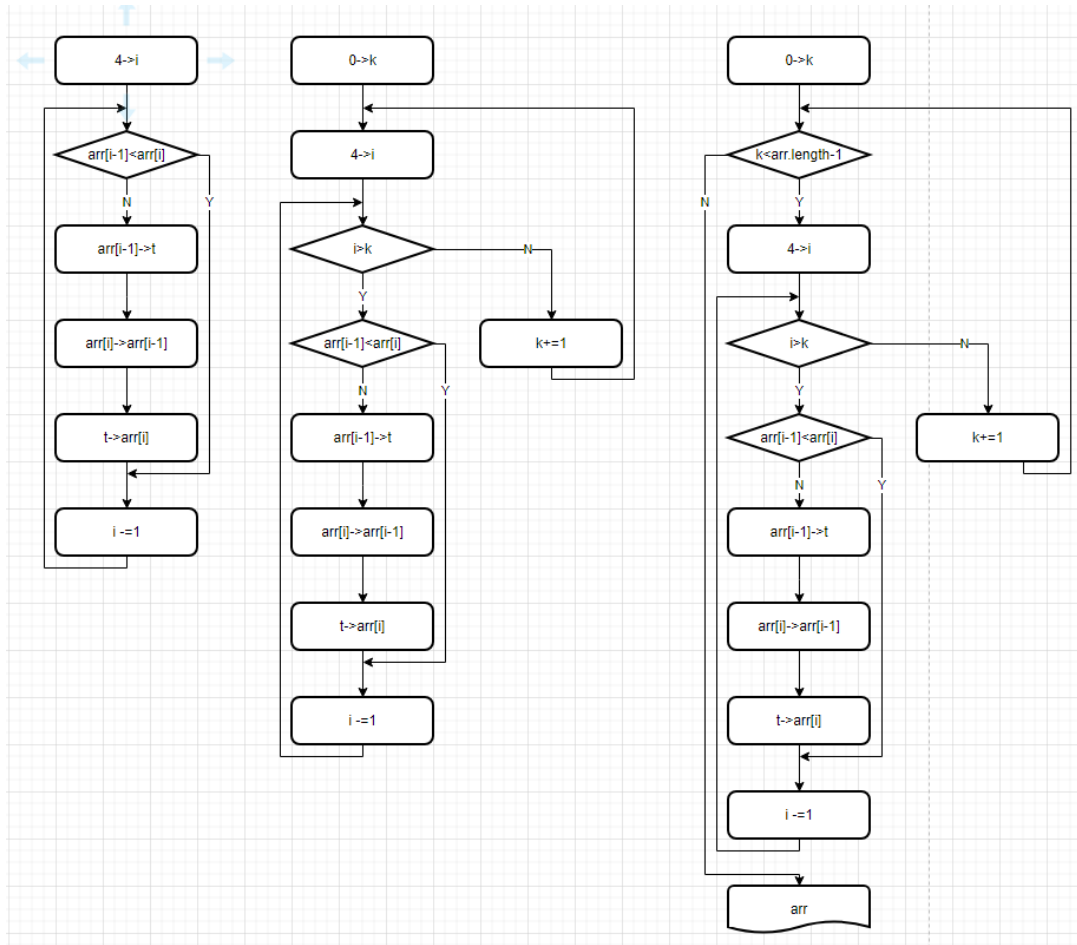
$arr[i-1] < arr[i]$



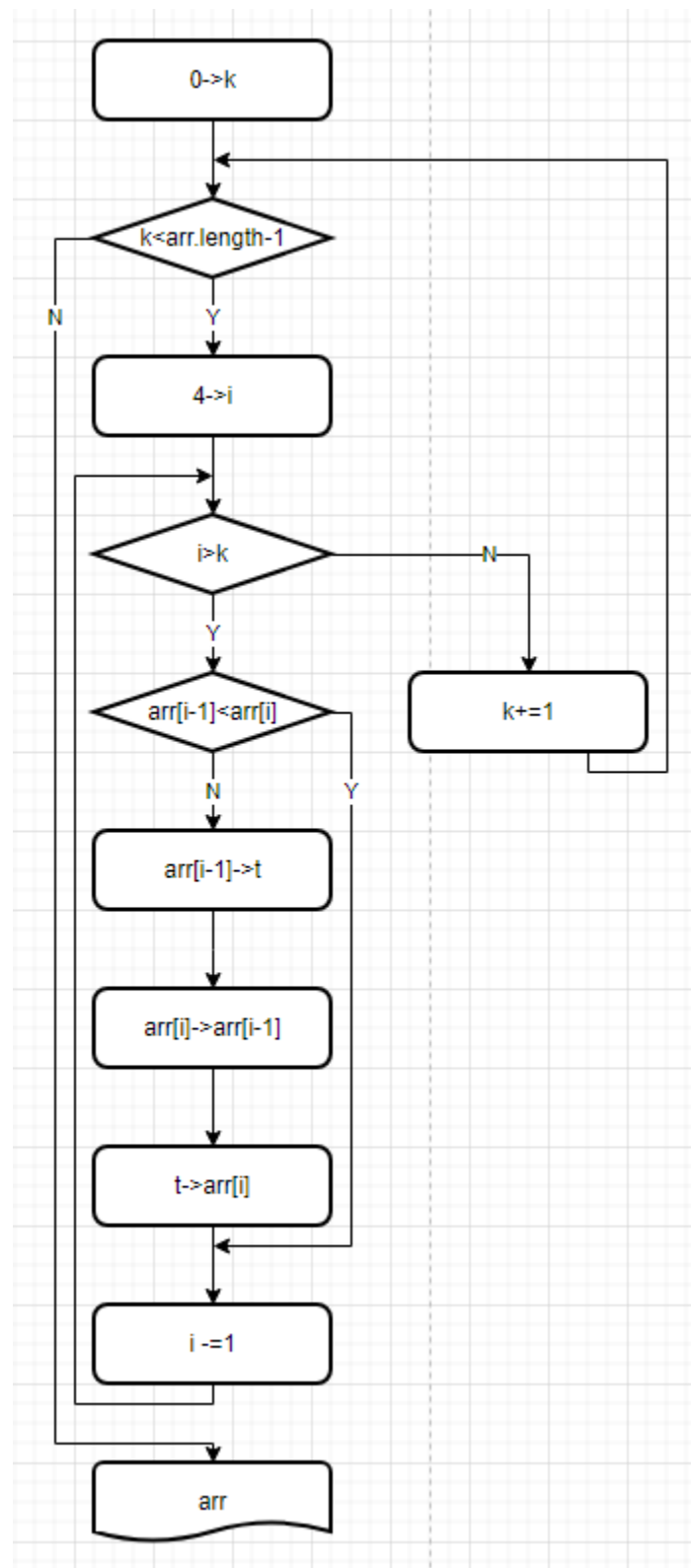
인접한 i, i-1요소의 데이터가 오름차순으로 정렬되어 있으면(yes) 교환을 건너뛰고

정렬되어있지 않으면(no) 교환이 이루어진다.





▼ Algorithm & Code



```

package day21;

public class day21_1 {
    public static void main(String[] args) {
        int[] arr = { 5, 3, 4, 1, 2 };
        int k = 0;
        int i = 4;

        for (k = 0; k < arr.length - 1; k += 1) {
            for (i = 4; i > k; i -= 1) {
                if (arr[i - 1] < arr[i]) {

                    } else {
                        int t = 0;
                        t = arr[i - 1];
                        arr[i - 1] = arr[i];
                        arr[i] = t;
                    }

                }

            }

            for (int x = 0; x < arr.length; x += 1) {
                System.out.print(arr[x] + " ");
            }

        }
    }
}

```