

# Selection Sort

📅 Date	
🕒 작성일시	@2022년 8월 3일 오전 10:29
▼ 강의 번호	
☰ 유형	
▼ 강사명	
<input checked="" type="checkbox"/> 강의자료	<input type="checkbox"/>
<input checked="" type="checkbox"/> 노션 복습	<input type="checkbox"/>
<input checked="" type="checkbox"/> 코딩 복습	<input type="checkbox"/>
<input checked="" type="checkbox"/> 주말숙제(교제)	<input type="checkbox"/>
<input checked="" type="checkbox"/> 정리	<input type="checkbox"/>

## ▼ 단순 선택법\_Selection Sort

- 데이터를 정렬하는 알고리즘 중에 하나이다.
- 가장 작은 데이터를 선택하여 맨 앞으로부터 순서대로 정렬한다.

### ▼ Algorithm

## Selection Sort

12	13	11	14	10
0	1	2	3	4

12	13	11	14	10
0	1	2	3	4

10	13	11	14	12
0	1	2	3	4

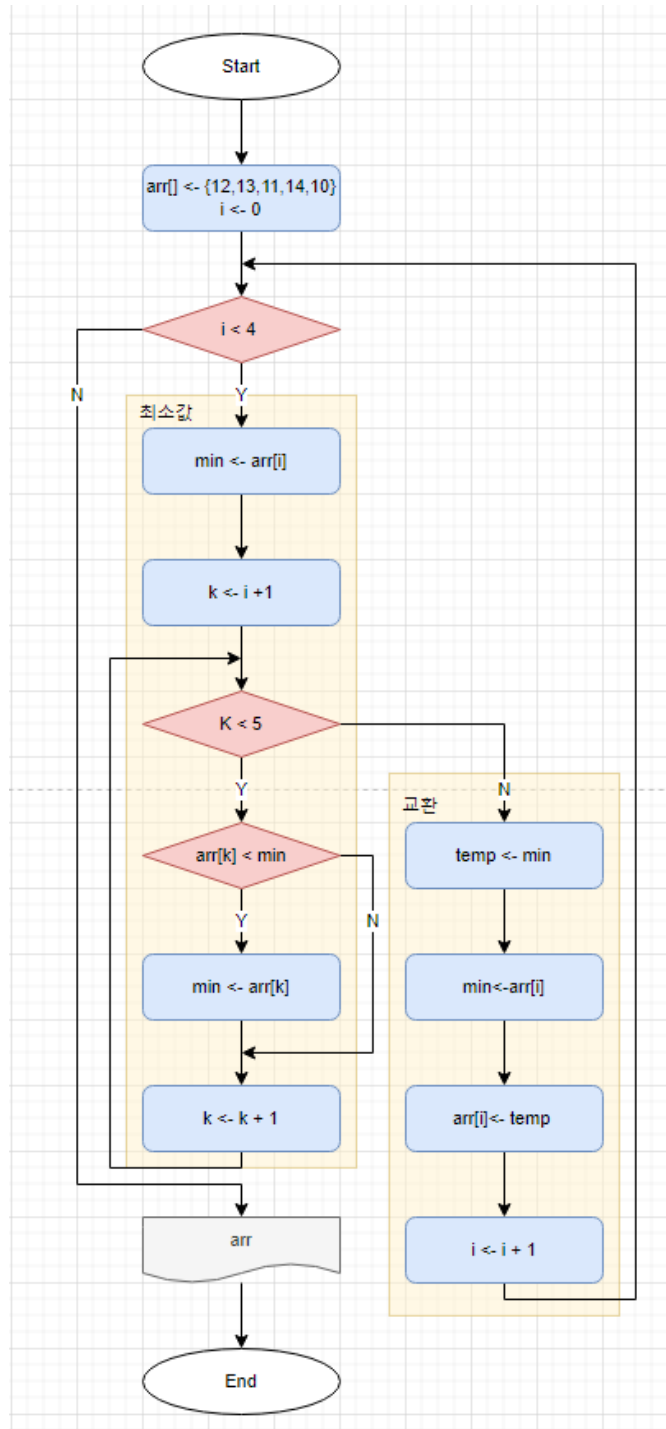
10	11	13	14	12
0	1	2	3	4

10	11	12	14	13
0	1	2	3	4

10	11	12	13	14
0	1	2	3	4

단순 선택법의 정렬 원리는 매우 간단하다.  
오름차순의 경우,  
먼저 가장 작은 숫자의 공을 찾는다.  
찾은 후에는 첫번째 공과 교환한다.  
이 동작을 반복하여 모든 공을 정렬하는 것이  
단순 선택 정렬의 기본적인 알고리즘이다.

### ▼ Flow chart



### ▼ Java code

```

public class SelectSort {

    public static void main(String[] args) {
        int[] arr = {12,13,11,14,10}; // 정렬할 배열
    }
}

```

```

System.out.print("정렬전 ");
for (int each : arr) {
    System.out.print(each + " ");
}
System.out.println();
System.out.println();

for (int i = 0; i < 4; i++) {
    int min = i; // 첫 번째값을 임시 최소값으로 가정

    for (int k = i + 1 ; k < 5; k++) {
        // 최소값 찾기 처리
        if (arr[k] < arr[min]) {
            min = k;
        }
    }
    // 최소값 변경이 필요할 경우 교환 처리
    int temp = arr[min];
    arr[min] = arr[i];
    arr[i] = temp;

    for (int each : arr) {
        System.out.print(each + " ");
    }
    System.out.println();
}
// 정렬된 최종 배열 출력
System.out.println();
System.out.print("정렬후 ");

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

Outputs :
정렬전 12 13 11 14 10

10 13 11 14 12
10 11 13 14 12
10 11 12 14 13
10 11 12 13 14

정렬후 10 11 12 13 14

```