

LinearSearch

🕒 작성일시	@2022년 8월 3일 오전 10:00
▼ 강의 번호	
▼ 유형	
🔗 자료	
≡ Property	
📅 Date	
☑ 복습	<input type="checkbox"/>
≡ 속성	

선형 탐색법 알고리즘

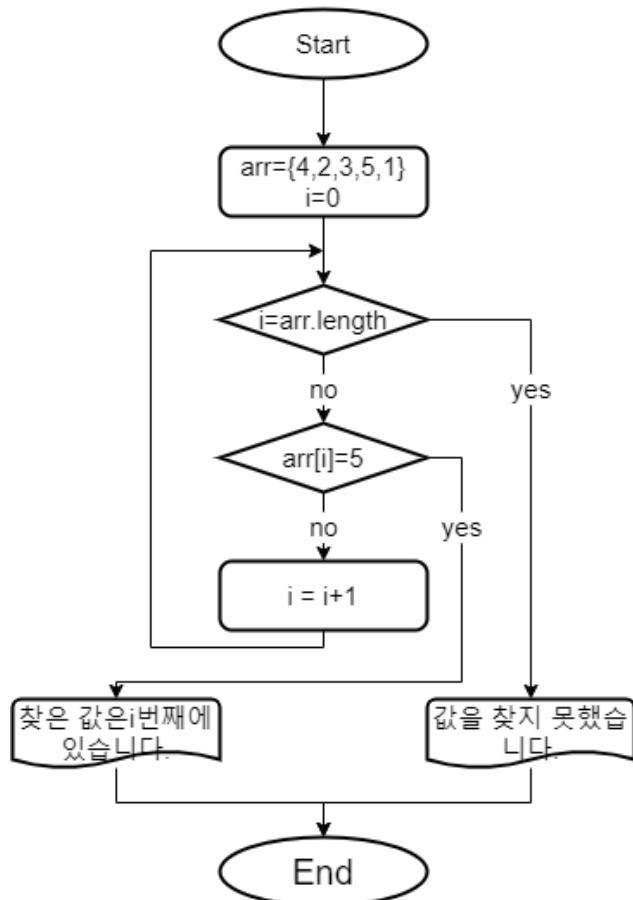
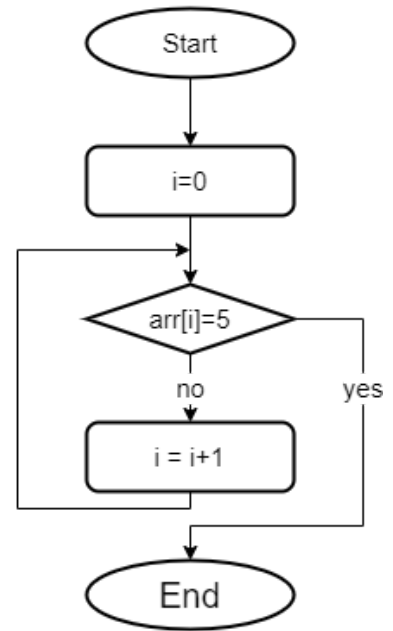
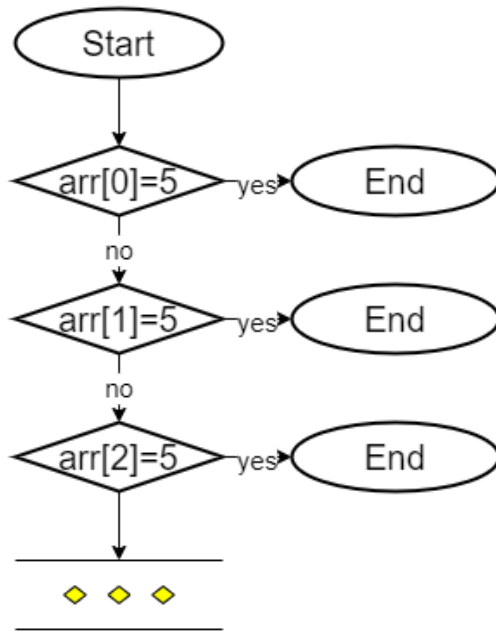
- 배열에 보관된 데이터를 맨 앞에서부터 순서대로 탐색하자
- 탐색처리는 반복 구조로 기술한다.
- 반복구조는 반드시 종료 조건이 중요하다.
-

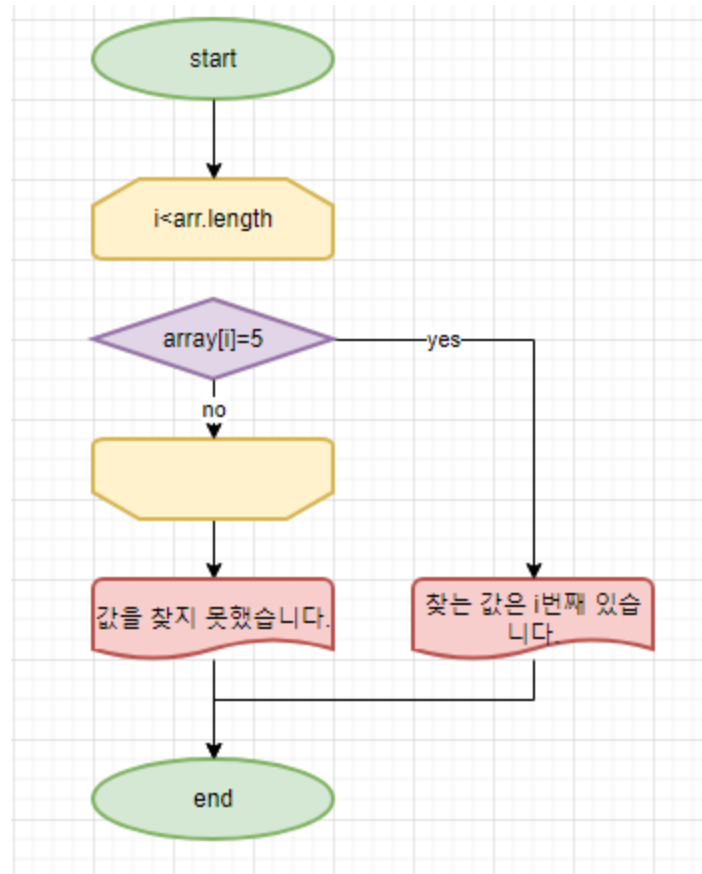
{4,2,3,5,1} ← 5를 찾는다

arr[0]=4 → arr[1]=2 → arr[2]=3 → arr[3]=5

▼ flowchart

Linear Search





▼ code

```

package day14;

public class day14_2 {
    public static void main(String[] args) {
        int arr[] = {4, 2, 3, 6, 1};
        int i = 0;
        for(i = 0; i < arr.length; i++) {
            if(arr[i] == 5) {
                System.out.println(arr[i] + "는 " + (i + 1) + "번째에 있습니다.");
                break;
            }
        }
        if(i == arr.length) {
            System.out.println("찾는 값이 없습니다.");
        }
    }
}
  
```

```

package day14;

public class day14_2_2 {
    public static void main(String[] args) {
        int arr[] = {4, 2, 3, 5, 1};
        int i = 0;
        int t = 0;
        for(i = 0; i < arr.length; i++) {
            if(arr[i] == 5) {
                t = 1;
                break;
            } else {t = 0;}
        }
        if(t == 1) {
            System.out.println(arr[i] + "는 " + (i + 1) + "번째에 있습니다.");
        } else {
            System.out.println("찾는 값이 없습니다.");
        }
    }
}

```