Search_Binary

작성일시	@2022년 8월 3일 오후 5:44
강의 번호	
유형	
강사	
자료	
복습	☐
날짜	@2022년 7월 26일

이진 탐색법 Binary Search

원하는 데이터를 찾는 알고리즘

반드시 찾는 데이터 전체가 정렬되어야만 사용 할 수 있다.(전제조건)

절반씩 대상 데이터를 줄여 가며 탐색한다.

알고리즘 순서

1. **가운데 요소를 찾는 처리 - 두 숫자의 평균구하기**
2. **가운데 요소와 원하는 데이터를 비교하는 처리**

center - 중간값

head

tail

(head + tail)/2 = center

요소들의 개수가 짝수 일때, 예를 들어 요소가 6이라 치면 center 후보가 두개가 된다. 하지만 2.5는 첨자가 될수가 없기 때문에 소수점 이하부분을 제거한 정수부분을 취하여 인덱스를 사용하면 전혀 문제가 없다.

평균 계산을 통해 가운데 요소의 값과 찾는 값을 비교하여 만약 첫방에 일치하면 프로그램을 종료하게 된다.

하지만 no의 경우, 즉 원하는 데이터가 아닐 경우에는 두가지 경우의 수 가 발생한다. 찾는값보다 작은 값인 경우와 찾는값보다 큰 값인 경우 이다. 이 두가지의 경우 탐색범위를 절반으로 줄이는 처리과정으로 넘어간다.

3. **탐색범위를 절반으로 줄이는 처리**

   3-1. 원하는 데이터가 가운데 데이터보다 큰경우 ex) arr[center] < 17

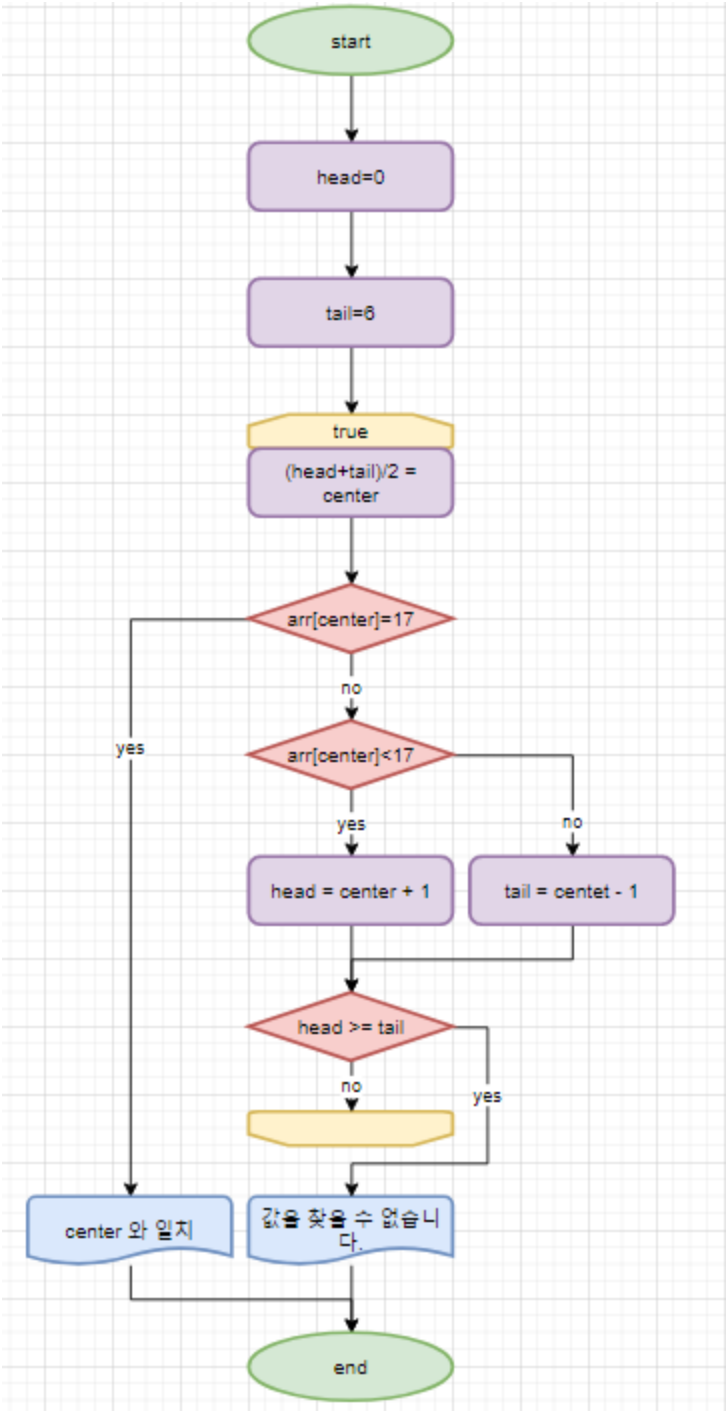
   이 경우 전체 검색범위의 뒷부분으로 대상을 절반으로 좁힌다. 따라서 탐색범위의 맨 앞 요소는 arr[center]보다 하나 큰 첨자를 갖는 요소가 된다.

   head = center + 1

   3-2. 원하는 데이터가 가운데 데이터 보다 작은 경우 ex) arr[center] > 17

   이 경우 전체 범위의 앞으로 대상을 절반으로 좁힌다. 따라서 탐색범위의 맨 뒷 요소는 arr[center]보다 하나 작은 첨자를 갖는 요소가 된다.

   tail = center - 1

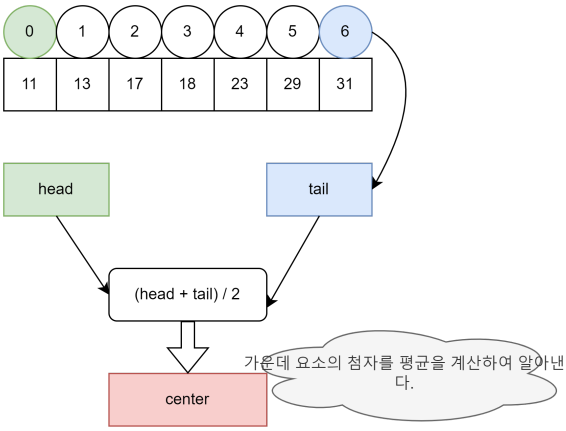


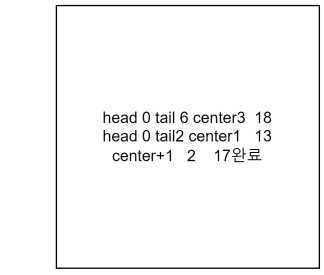
4. 가운데 요소를 선택처리 - 평균 사용
5. 가운데 데이터와 원하는 값을 비교 처리

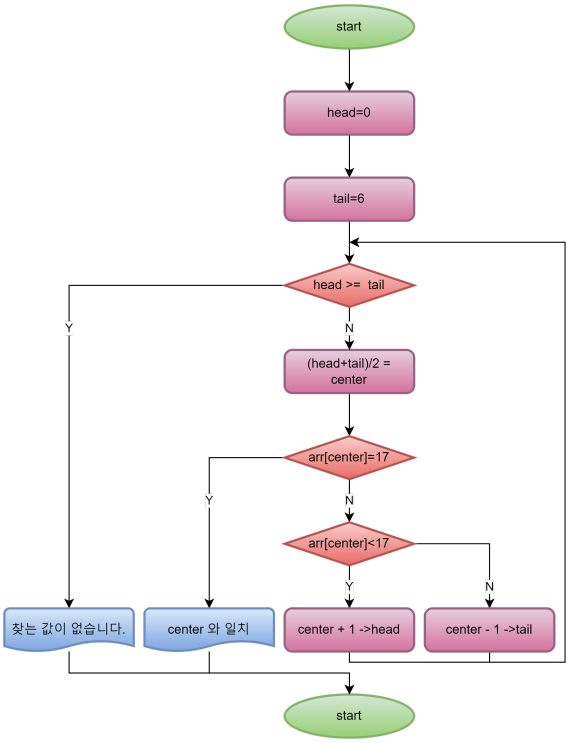
6. 탐색범위를 반으로 줄이는 처리 - 찾은 중간값이 원하는 값보다 작으면 head ← center + 1

찾은 중간값이 원하는 값보다 크면 tail ← center - 1

검색종료지점 (탈출지점) 조건은 head >= tail








```
package algo;

public class Binary {

  public static void main(String[] args) {
    int[] arr = {11, 13, 17, 18, 23, 29, 31};
    int ans = 17;

    int result = bsearch(arr, ans);

    if (result != -1) {
      System.out.println(result + "번째 요소 일치");
    }else {
      System.out.println("찾는 값이 없습니다.");
    }
  }
```

```
  public static int bsearch(int[] arr, int ans) {
    int head = 0;
    int tail = 6;
    while(head<=tail) {
      int center = (head+tail)/2;
      if(arr[center] == ans) {
        return center;
      } else if(arr[center] < ans) {
        head = center + 1;
      } else {
        tail = center - 1;
      }
    }
    return -1;
  }


}
```