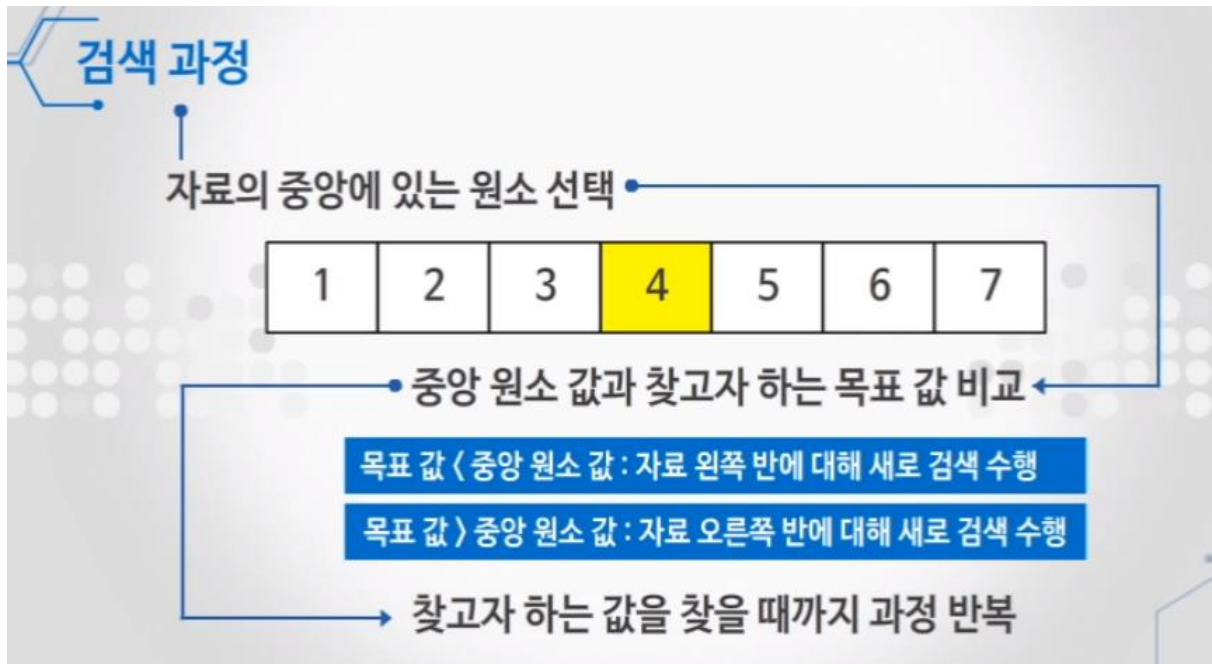


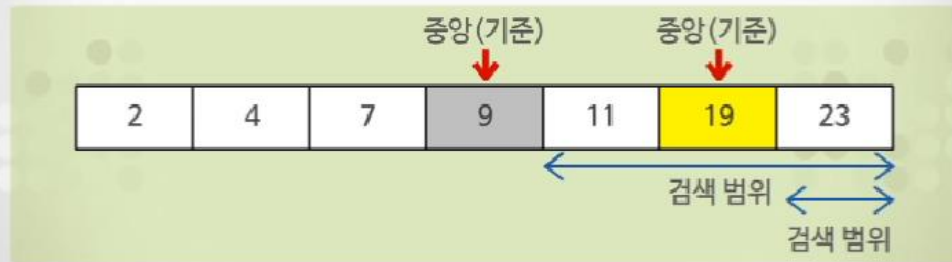
이진검색

- 정렬되어 있는 자료의 가운데에 있는 항목의 키 값과 비교하여 다음 검색의 위치를 결정하고 검색을 계속 진행하는 방법
 - 검색 범위를 반으로 줄여가면서 목적 값을 찾을 때까지 순환적으로 반복 수행



검색 과정

예 이진 검색으로 23 찾기



01 가운데 원소 9를 기준으로 선택

03 가운데 원소 19를 기준으로 선택

05 $20 \neq 23 \Rightarrow$ 검색 실패

02 $20 > 9 \Rightarrow$ 오른쪽 검색

04 $20 > 19 \Rightarrow$ 오른쪽 검색

```

2  자료에 삽입, 삭제 발생 시 배열의 상태를 항상 정렬 상태로 유지하는 추가 작업이 필요
3  //반복문
4  ▾ binarySearch( n, S[], key){
5      low <- 0
6      high <- n + 1
7
8  ▾ while( low <= high && location = 0){
9      mid <- low + (high - low) / 2
10 ▾ if( S[mid] == key)
11     return mid
12 ▾ else if(S[mid] > key)
13     high <- mid - 1
14 ▾ else
15     low <- mid + 1
16     return -1
17 }
18 }
19
20
21 //재귀
< 2 ▾ binarySearch(S[], low, high, key){
23 ▾ if(low > high)
24     return -1
25 ▾ else {
26     mid <- (low + high) / 2
27
28 ▾ if(key == S[mid])
29     return mid
30 ▾ else if(key < a[mid])
31     return binarySearch(A[], low, mid-1, key)
32 ▾ else {
33     return binarySearch(A[], mid+1, high, key)
34 }
35 }
36 }

```

분할 정복 사례

병합 정렬

- 외부 정렬의 기본이 되는 정렬 알고리즘
- 활용 : Multi-Core CPU, 다수의 프로세서에서 정렬 알고리즘의 병렬화

퀵 정렬

- 매우 큰 입력 데이터에 대해 좋은 성능을 보임
- 활용 : 생명 정보 공학(Bioinformatics)에서 특정 유전자를 효율적으로 찾는데 접미어 배열과 함께 사용

최근접 점의 쌍(Closest Pair) 문제에 분할 정복 기법 적용 가능

2차원 평면상의 n 개의 점이 입력으로 주어질 때, 거리가 가장 가까운 한 쌍의 점을 찾는 문제
활용 : 컴퓨터 그래픽스, 컴퓨터 비전, 지리 정보 시스템, 항공 트래픽 제어, 마케팅 등