
검색(1)

순차검색

❖ 순차 검색 sequential search, 선형 검색 linear search

- 일렬로 된 자료를 처음부터 마지막까지 순서대로 검색하는 방법
- 가장 간단하고 직접적인 검색 방법
- 배열이나 연결 리스트로 구현된 순차 자료 구조에서 원하는 항목을 찾는 방법
- 검색 대상 자료가 많은 경우에 비효율적이지만 알고리즘이 단순하여 구현이 용이함

순차검색

❖ 정렬되어 있지 않은 자료를 순차 검색

○ 검색 방법

- 첫 번째 원소부터 시작하여 마지막 원소까지 순서대로 키 값이 일치하는 원소가 있는지를 비교하여 찾는다.
- 키 값이 일치하는 원소를 찾으면 그 원소가 몇 번째 원소인지를 반환
- 마지막 원소까지 비교하여 키 값이 일치하는 원소가 없으면 찾은 원소가 없는 것이므로 검색 실패

순차검색

8	30	1	9	11	19	2
---	----	---	---	----	----	---

(a) 정렬되어 있지 않은 자료의 예

① $8 \neq 9$	8	30	1	9	11	19	2
② $30 \neq 9$	8	30	1	9	11	19	2
③ $1 \neq 9$	8	30	1	9	11	19	2
④ $9 = 9$	8	30	1	9	11	19	2

순차검색

① $8 \neq 6$	8	30	1	9	11	19	2
② $30 \neq 6$	8	30	1	9	11	19	2
③ $1 \neq 6$	8	30	1	9	11	19	2
④ $9 \neq 6$	8	30	1	9	11	19	2
⑤ $11 \neq 6$	8	30	1	9	11	19	2
⑥ $19 \neq 6$	8	30	1	9	11	19	2
⑦ $2 \neq 6$	8	30	1	9	11	19	2

(c) 검색 실패의 예 : 6 검색

그림 10-1 정렬되어 있지 않은 자료에서의 순차 검색 예

순차검색

❖ 정렬된 자료를 순차 검색

- 원소의 키값이 찾는 키값보다 크면 찾는 원소가 없는 것이므로 더 이상 검색을 수행하지 않아도 검색 실패를 알 수 있음

8	30	1	9	11	19	2
---	----	---	---	----	----	---

(a) 정렬되어 있지 않은 자료의 예

① $8 \neq 9$	8	30	1	9	11	19	2
② $30 \neq 9$	8	30	1	9	11	19	2
③ $1 \neq 9$	8	30	1	9	11	19	2
④ $9 = 9$	8	30	1	9	11	19	2

순차검색

① $1 < 6$	1	2	8	9	11	19	29
② $2 < 6$	1	2	8	9	11	19	29
③ $8 > 6$	1	2	8	9	11	19	29

→ 검색 종료

(c) 검색 실패의 예 : 6 검색

그림 10-2 정렬된 자료에서의 순차 검색 예

이진 검색

❖ 이진 검색binary search, 이분 검색, 보간 검색interpolation search

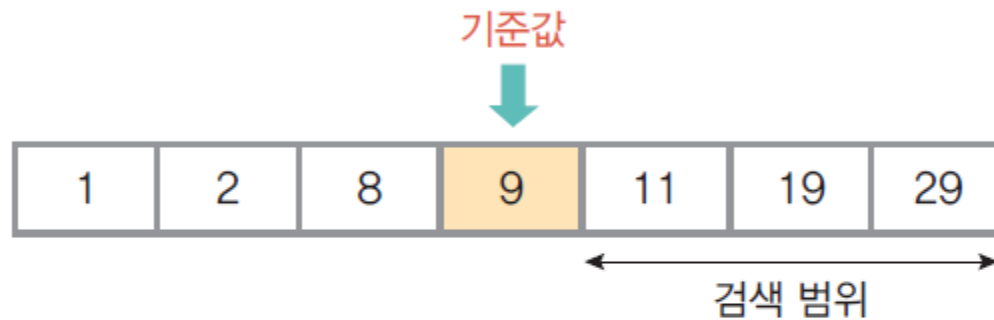
- 자료의 가운데에 있는 항목을 키 값과 비교하여 다음 검색 위치를 결정하여 검색을 계속하는 방법
 - 찾는 키 값 > 원소의 키 값 : 오른쪽 부분에 대해서 검색 실행
 - 찾는 키 값 < 원소의 키 값 : 왼쪽 부분에 대해서 검색 실행
- 키를 찾을 때까지 이진 검색을 순환적으로 반복 수행함으로써 검색 범위를 반으로 줄여가면서 더 빠르게 검색
- 정복 기법을 이용한 검색 방법
 - 검색 범위를 반으로 분할하는 작업과 검색 작업을 반복 수행
- 정렬되어있는 자료에 대해서 수행하는 검색 방법

이진 검색

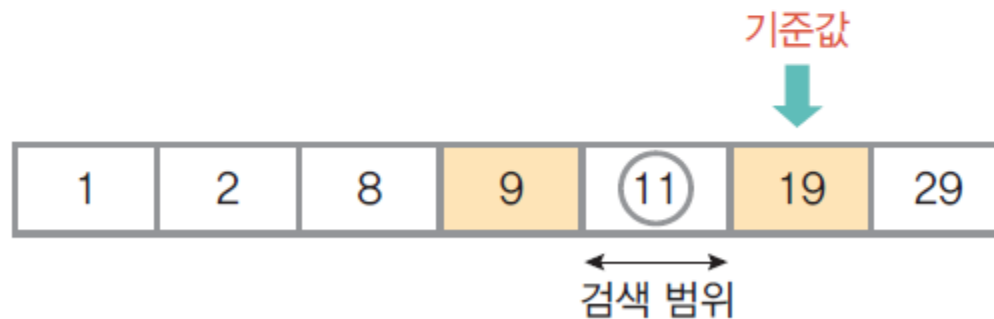
1	2	8	9	11	19	29
---	---	---	---	----	----	----

(a) 이진 검색 자료 예

① $11 > 9 \rightarrow$ 오른쪽 종료



② $11 < 19 \rightarrow$ 왼쪽 종료

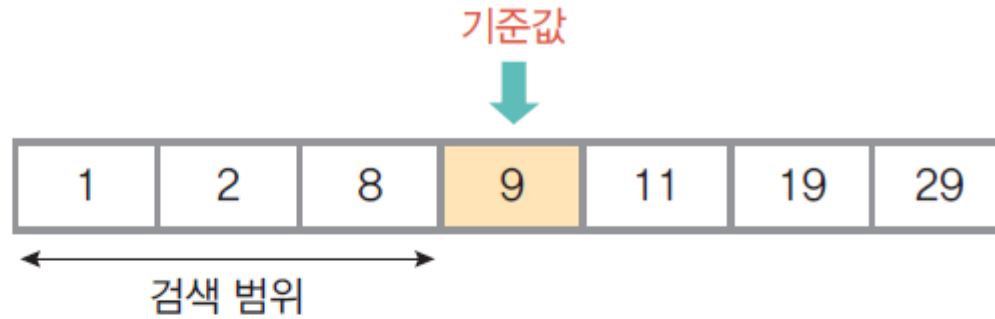


③ $11 = 11 \rightarrow$ 검색 성공

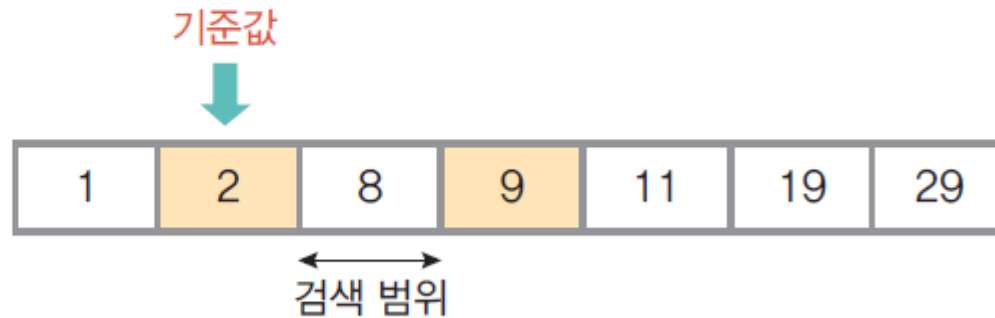
(b) 검색 성공의 예 : 11 검색

이진 검색

① $6 < 9 \rightarrow$ 왼쪽 종료



② $6 > 2 \rightarrow$ 오른쪽 종료



③ $6 \neq 8 \rightarrow$ 검색 종료

(c) 검색 실패의 예 : 6 검색

그림 10-4 이진 검색의 예

이진 검색

❖ 이진 검색 소스

```
public int search(int[] arr, int target) {
    int first = 0;
    int last = arr.length;
    int mid;

    while (first <= last) { // 검색 종료 조건
        mid = (first + last) / 2;
        if (target == arr[mid]) { // 찾음
            return mid;          // 검색된 인덱스 리턴
        } else {
            if (target < arr[mid]) {
                last = mid - 1;
            } else {
                first = mid + 1;
            }
        }
    }
    return -1; // 검색 실패, target이 없음
}
```