

14주차 1차시 검색과 순차 검색

【학습목표】

1. 자료에 대한 검색의 개념을 설명할 수 있다.
2. 순차 검색의 개념과 알고리즘을 설명할 수 있다.

학습내용1 : 검색과 검색 방법

1. 검색

- * 컴퓨터에 저장한 자료 중에서 원하는 항목을 찾는 작업
 - 검색 성공 - 원하는 항목을 찾은 경우
 - 검색 실패 - 원하는 항목을 찾지 못한 경우
- * 탐색 키를 가진 항목을 찾는 것
 - 탐색 키(search key) - 자료를 구별하여 인식할 수 있는 키
- * 삽입/삭제 작업에서의 검색
 - 원소를 삽입하거나 삭제할 위치를 찾기 위해서 검색 연산 수행

2. 검색 방법

- * 수행 위치에 따른 분류
 - 내부 검색 - 메모리 내의 자료에 대해서 검색 수행
 - 외부 검색 - 보조 기억 장치에 있는 자료에 대해서 검색 수행
- * 검색 방식에 따른 분류
 - 비교 검색 방식(comparison search method)
 - 검색 대상의 키를 비교하여 검색하는 방법
 - 순차 검색, 이진 검색, 트리 검색
 - 계산 검색 방식(non-comparison method)
 - 계수적인 성질을 이용한 계산으로 검색 하는 방법
 - 해싱
- * 검색 방법의 선택
 - 자료 구조의 형태와 자료의 배열 상태에 따라 최적의 검색 방법 선택

학습내용2 : 정렬, 비정렬 순차 검색

1. 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

2. 정렬되지 않은 순차 자료구조에서의 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
 - 순차 검색 성공의 경우

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

9를 검색하는 경우

① 8 ≠ 9

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

② 30 ≠ 9

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

③ 1 ≠ 9

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

④ 9 = 9

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

 (a) 검색 성공의 경우

- 순차 검색 패의 경우

6을 검색하는 경우

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
---	----	---	---	----	----	---

 (b) 검색 실패의 경우

[그림 11-1] 순차 검색의 예

* 정렬되어있지 않은 자료에 대한 순차 검색 알고리즘

알고리즘 11-1 순차 검색 알고리즘 1

```

sequentialSearch1(a[], n, key)
    i ← 0;
    while (i < n and a[i] ≠ key) do {
        i ← i + 1;
    }
    if (i < n) then return i;
    else return -1;
end sequentialSearch1()
    
```

- 비교횟수 : 찾고자 하는 원소의 위치에 따라 결정

- 찾는 원소가 첫 번째 원소라면 비교횟수는 1번, 두 번째 원소라면 비교횟수는 2번, 세 번째 원소라면 비교횟수는 3번, 찾는 원소가 i번째 원소이면 i번,

- 정렬되지 않은 원소에서의 순차 검색의 평균 비교 횟수

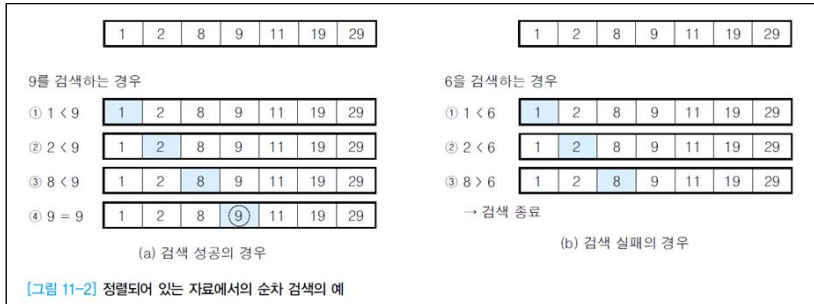
$$= 1/n(1+2+3+ \dots + n) = (n+1)/2$$

- 평균 시간 복잡도 : $O(n)$

3. 정렬되어 있는 순차 자료구조에서의 순차 검색

* 검색 방법

- 순서대로 검색하면서 키 값을 비교하여, 원소의 키 값이 찾는 키 값보다 크면 찾는 원소가 없는 것이므로 더 이상 검색을 수행하지 않고 검색종료
- 정렬되어있는 자료에 대한 순차 검색 예)



* 정렬되어있는 자료에 대한 순차 검색 알고리즘

알고리즘 11-2 순차 검색 알고리즘 2

```

sequentialSearch2(a[], n, key)
    i ← 0;
    while (a[i] < key) do {
        i ← i + 1;
    }
    if (a[i] = key) then return i;
    else return -1;
end sequentialSearch2()

```

- 비교횟수 - 찾고자 하는 원소의 위치에 따라 결정
 - 검색 실패의 경우에 평균 비교 횟수가 반으로 줄어든다.
 - 정렬되어있는 원소에서의 순차 검색의 평균 비교 횟수

$$= 1/n(1+2+3+ \dots + n) \times 1/2 = (n+1)/4$$
- 평균 시간 복잡도 : $O(n)$

학습내용3 : 색인 순차 검색

1. 정의

* 정렬되어있는 자료에 대한 인덱스 테이블(index table)을 추가로 사용하여 탐색 효율을 높인 검색 방법

* 인덱스 테이블

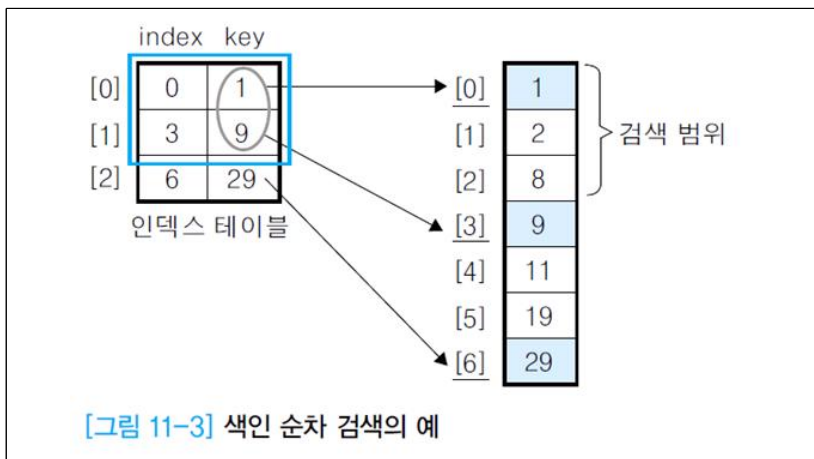
- 배열에 정렬되어있는 자료 중에서 일정한 간격으로 떨어져있는 원소들을 저장한 테이블
 - 자료가 저장되어있는 배열의 크기가 n 이고 인덱스 테이블의 크기가 m 일 때, 배열에서 n/m 간격으로 떨어져있는 원소와 그의 인덱스를 인덱스 테이블에 저장

* 검색 방법

- $\text{indexTable}[i].\text{key} \leq \text{key} < \text{indexTable}[i+1].\text{key}$ 를 만족하는 i 를 찾아서 배열의 어느 범위에 있는지를 먼저 알아낸 후에 해당 범위에 대해서만 순차 검색 수행

* 색인 순차 검색 예제

- 검색 대상 자료 : {1, 2, 8, 9, 11, 19, 29 }
- 크기가 3인 인덱스 테이블 작성
- 인덱스 테이블에서 먼저 탐색키를 검색하여 검색 범위를 확인하고, 해당 범위에 대해서만 순차 검색 실행



* 색인 순차 검색의 성능

- 인덱스 테이블의 크기에 따라 결정
 - 인덱스 테이블의 크기가 줄어들면 배열의 인덱스를 저장하는 간격이 커지므로 배열에서 검색해야하는 범위도 커진다.
 - 인덱스 테이블의 크기를 늘리면 배열의 인덱스를 저장하는 간격이 작아지므로 배열에서 검색해야하는 범위는 작아지겠지만 인덱스 테이블을 검색하는 시간이 늘어나게 된다.
- 색인 순차 검색의 시간 복잡도 : $O(m + n/m)$
 - 배열의 크기 : n , 인덱스 테이블의 크기 : m

【학습정리】

1. 자료를 검색하는 것은 원하는 탐색기를 가진 항목을 찾는 것이다. 원하는 항목을 찾은 경우를 검색 성공이라 하고 찾지 못하는 경우는 검색 실패라고 한다.

2. 검색 방법에 따른 분류

- 비교 검색 방식 : 검색 대상의 키를 비교하여 검색하는 방법으로 순차 검색, 이진 검색, 트리 검색이 있다.
- 계산 검색 방식 : 키를 비교하지 않고 계수적인 성질을 이용한 계산으로 검색하는 방법으로 해싱이 있다.