

실습10 - Searching

실습10 탐색(Searching)

실습 목적 및 탐색(Searching)의 이해

■ 실습 목적

- 여러 가지 탐색 방법들을 이해한다.
- 다양한 탐색 알고리즘을 구현해보고, 각 탐색 방법의 효율을 확인해 본다.

■ 탐색(search)은 기본적으로 여러 개의 자료 중에서 원하는 자료를 찾는 작업

- **순차 탐색(sequential search)**: 탐색 방법 중에서 가장 간단하고 직접적인 탐색 방법
- **이진 탐색(binary search)**: 배열의 중앙에 있는 값을 조사하여 찾고자 하는 항목이 왼쪽 또는 오른쪽 부분 배열에 있는지를 알아내어 탐색의 범위를 반으로 줄이는 탐색 방법
- **색인 순차 탐색(indexed sequential search)**: 인덱스(index) 테이블을 사용하여 탐색의 효율을 높이는 방법
- **보간탐색(interpolation search)**: 사전이나 전화번호부를 탐색하는 방법과 같이 탐색키가 존재할 위치를 예측하여 탐색하는 방법

실습 목적 및 탐색(Searching)의 이해

■ 순차 탐색(sequential search)

• 8을 찾는 경우

(1) $9 \neq 8$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

(2) $5 \neq 8$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

(3) $8 = 8$ 이므로 탐색 성공

9	5	8	3	7
---	---	---	---	---

(a) 탐색 성공의 경우

• 2를 찾는 경우

(1) $9 \neq 2$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

(2) $5 \neq 2$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

(3) $8 \neq 2$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

(4) $3 \neq 2$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

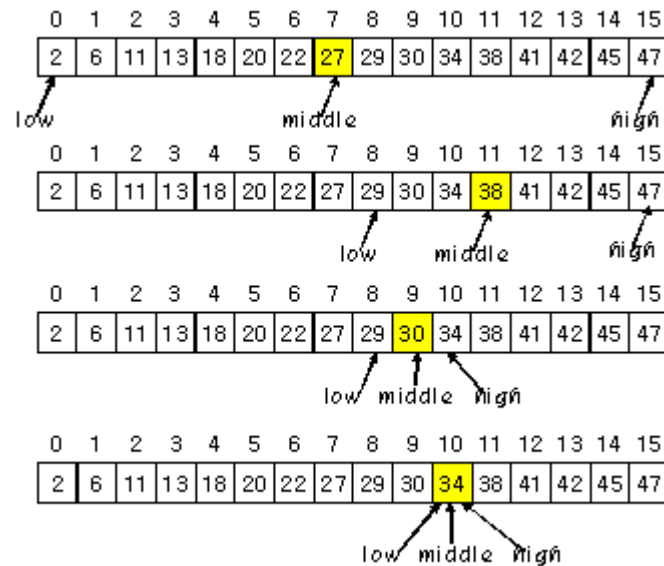
(5) $7 \neq 2$ 이므로 탐색 계속

9	5	8	3	7
---	---	---	---	---

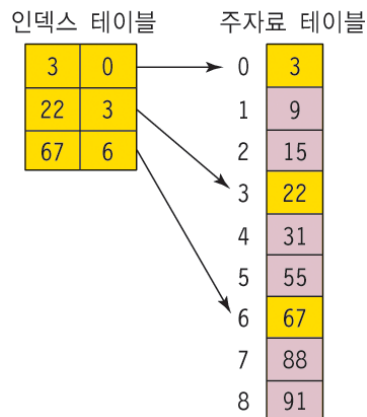
(6) 더 이상 항목이 없으므로
탐색 실패

(b) 탐색 실패의 경우

■ 이진 탐색(binary search)

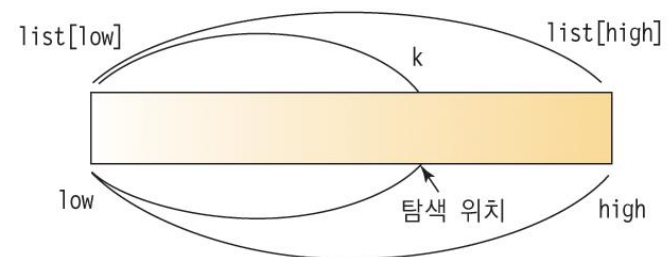


■ 색인 순차 탐색(indexed sequential search)



■ 보간탐색(interpolation search)

$(list[high] - list[low]) : (k - list[low]) = (high - low) : \text{탐색 위치} - low$



실습 문제 - 탐색(Searching)

■ 실습 프로그램 설명

- 각 탐색 방법의 탐색 시간과 문자 수행횟수 결과를 출력해본다.
- 다른 탐색 방법과 비교해본다.

■ 주어진 함수

- 탐색 함수
 - `int seqsearch(int list[], int n, int searchnum)`
 - `int seqsearch2(int list[], int n, int searchnum)`
 - `int seqsearch3(int low, int high, int searchnum)`
 - `int binsearch(int list[], int n, int searchnum)`
 - `int binsearch2(int list[], int low, int high, int searchnum)`
 - `int index_search(int list[], int key, int n)`
 - `int search_interpolation(int list[], int n, int key)`

■ 기타 함수

- `void makeIndexTable(int list[], int size)`
 - 인덱스 테이블 생성하는 함수
- 탐색에 대한 설명은 Chapter11.Searching 강의노트 참조

실습 문제 - 탐색

■ 실습 문제 : 탐색의 구현 및 사용

- 다양한 탐색 알고리즘을 구현해 본다.
- 탐색의 방법을 다른 탐색 방식과 비교한다.

■ 메인 함수 설명

- 각 탐색에 대하여, 각 탐색 함수의 수행 횟수와 탐색 시간을 확인해볼 수 있다.
- 탐색한 숫자의 위치를 출력한다.

■ 결과 화면(예)

```
C:\WINDOWS\system32\cmd.exe
탐색할 숫자를 입력하세요: 4999999
순차 탐색시간 = 30 milliseconds
문자의 수행횟수 = 5000000
숫자가 위치 4999999에서 발견되었습니다.
개선된 순차 탐색시간 = 26 milliseconds
문자의 수행횟수 = 4999999
숫자가 위치 4999999에서 발견되었습니다.
이진 탐색(반복 버전) 시간 = 0 milliseconds
문자의 수행횟수=1
숫자가 위치 4999999에서 발견되었습니다.
이진 탐색(순환 버전) 시간 = 0 milliseconds
문자의 수행횟수=1
숫자가 위치 4999999에서 발견되었습니다.
색인 순차 탐색시간 = 0 milliseconds
문자의 수행횟수=38999
숫자가 위치 4999999에서 발견되었습니다.
보간 탐색시간 = 0 milliseconds
문자의 수행횟수 = 1
숫자가 위치 4999999에서 발견되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\WINDOWS\system32\cmd.exe
탐색할 숫자를 입력하세요: 2499999
순차 탐색시간 = 17 milliseconds
수행횟수 = 2500000
숫자가 위치 2499999에서 발견되었습니다.
개선된 순차 탐색시간 = 16 milliseconds
수행횟수 = 2499999
숫자가 위치 2499999에서 발견되었습니다.
이진 탐색(반복 버전) 시간 = 0 milliseconds
수행횟수=2
숫자가 위치 2499999에서 발견되었습니다.
이진 탐색(순환 버전) 시간 = 0 milliseconds
수행횟수=2
숫자가 위치 2499999에서 발견되었습니다.
색인 순차 탐색시간 = 1 milliseconds
수행횟수=39031
숫자가 위치 2499999에서 발견되었습니다.
보간 탐색시간 = 0 milliseconds
수행횟수 = 1
숫자가 위치 2499999에서 발견되었습니다.
계속하려면 아무 키나 누르십시오 . . .
```

Thank You
Q&A