1 레코드, 키의 정의 및 탐색 트리

1 레코드, 키의 정의 및 탐색 트리

1 탐색(검색)

- 컴퓨터에서 자료를 찾는 방법
- 기억 공간에 보관 중인 데이터 중에서 원하는 정보를 찾아내는 작업
- 컴퓨터 안에는 엄청나게 많은 자료들이 있는데 컴퓨터는 자료를 빨리 찾을 수 있도록 일정한 논리 순서에 맞추어 작업을 하는데 이때 필요한 논리 순서가 탐색 알고리즘임

1 레코드, 키의 정의 및 탐색 트리

1 탐색(검색)

- 책에서 어떤 내용을 찾기 위해 페이지를 마구 뒤지는 일

  ↳ 색인(Index)이 있으면 편리하게 책의 해당 페이지를 찾을 수 있음

- 엄청난 양의 웹 문서들을 빠른 시간에 검색해주는 구글과 같은 검색 엔진이 대표적인 탐색의 응용 예

- 탐색의 종류로는 순차 탐색, 이진 탐색 등이 있음

1 레코드, 키의 정의 및 탐색 트리

1 탐색(검색)

적절한 자료구조와 알고리즘의 사용은
효율적인 데이터의 저장과 탐색에서 매우 중요함

- 데이터의 저장과 검색은 자료구조와 알고리즘에서 매우 중요
- 데이터를 저장하는 효율적 자료구조가 개발되기 전에는 데이터가 들어오는 대로 쌓는 방법밖에 없었음

1 레코드, 키의 정의 및 탐색 트리

1 탐색(검색)

- 배열에 데이터가 들어오는 순서대로 쌓는 것은 데이터를 저장하기는 쉽지만 검색은 번거로움

➔ 이진 탐색 트리와 해시 테이블은 자료를 찾는 색인 역할을 하는 자료구조

2 레코드, 키, 탐색트리

레코드(Record)

- 개체에 대해 수집된 모든 정보를 포함하고 있는 저장 단위
  - 예 : 사람의 레코드라면 주민번호, 이름, 집주소, 연락처, 최종 학력, 이메일 등의 정보 포함

[사람]

주민번호	이름	집주소	연락처	최종 학력	이메일

2 레코드, 키, 탐색트리

필드(Field)

- 레코드에서 각각의 정보를 나타내는 부분
  - 예 : 사람의 레코드에서 각각의 정보를 나타내는 부분

2 레코드, 키, 탐색트리

탐색 키(Search key) 또는 키(Key)

- 다른 레코드와 중복되지 않도록 각 레코드를 대표할 수 있는 필드
  - 예 : 사람 레코드의 경우 주민번호만 있으면 다른 사람 레코드와 구분될 수 있으며 해당 레코드를 대표할 수 있으므로 주민번호가 탐색 키가 될 수 있음

2 레코드, 키, 탐색트리

탐색 키(Search key) 또는 키(Key)

- 키는 필드 하나로 구성할 수도 있고, 두 개 이상의 필드로 구성할 수도 있음
  - 예 : 이름과 연락처를 사용하면 두 개의 필드로 구성된 키가 될 수 있음

2 레코드, 키, 탐색트리

탐색 트리(Search Tree)

- 각 노드가 규칙에 맞도록 하나씩의 키를 갖고 있음
- 이를 통해 해당 레코드가 저장된 위치를 알 수 있음
- 저장되는 장소에 따라 내부 탐색 트리와
  외부 탐색 트리로 나뉨

2 레코드, 키, 탐색트리

탐색 트리(Search Tree)

내부 탐색 트리

- 탐색 트리가 메인 메모리 내에 존재
- 메인 메모리에서 모든 키를 수용할 수 있으면
  탐색 트리 전체를 메인 메모리로 한번만 탑재한 후
  내부 탐색 트리로 사용할 수 있음

2 레코드, 키, 탐색트리

탐색 트리(Search Tree)

외부 탐색 트리

- 탐색 트리가 외부(주로 디스크)에 존재
- 메인 메모리에서 모든 키를 수용할 수 없을 정도로 크면 디스크 공간에 저장된 상태로 탐색해야 함

2 순차 탐색

2 순차 탐색

1 순차 탐색(Sequential Search)

- 이해하기 쉽고 간단한 탐색 알고리즘
- 선형 탐색(Linear Search)이라고도 함
- 순서화되어 있지 않은 경우 사용하며
  첫 번째 데이터부터 차례로 비교하여 탐색하는 방식
- 원하는 데이터를 찾을 때까지 키를 하나씩
  비교해가는 방식

2 순차 탐색

1 순차 탐색(Sequential Search)

- 데이터가 정렬되어 있을 때도 하나씩 비교하면서 원하는 값을 찾기 때문에 비효율적
- 파일이 크면 탐색 시간 증가
  - 예 : 무작위로 모여진 명함 중에서 원하는 사람의 명함을 찾는 경우
- 시간 복잡도: O(n)

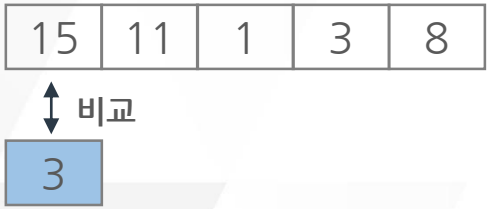
2 순차 탐색 방법

◆ 예) 데이터 집합에서 순차 탐색을 이용해 데이터 3을 탐색하는 과정

[데이터 집합]

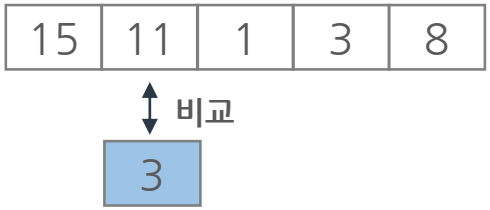
15	11	1	3	8

① 첫 번째 데이터인 15와 찾고자 하는 3이 같은지 비교하는데 다르므로 다음으로 이동



2 순차 탐색 방법

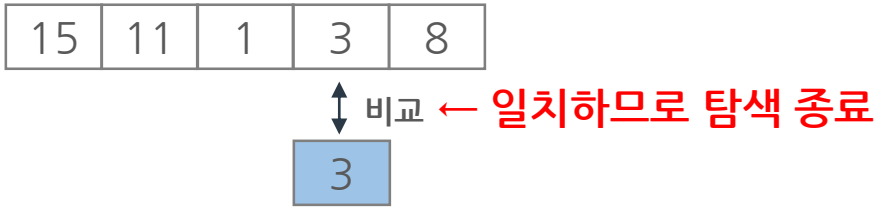
② 두 번째 데이터인 11과 찾고자 하는 3이 같은지 비교하여 다르므로 다음으로 이동



③ 세 번째 데이터인 1과 찾고자 하는 3이 같은지 비교하여 다르므로 다음으로 이동

2 순차 탐색 방법

④ 네 번째 데이터인 3과 찾고자 하는 3이 같은지 비교하여 같으므로 원하는 데이터를 찾고 탐색을 종료함



만약 마지막까지 원하는 데이터를 찾지 못하면 탐색 실패로 종료함

3 자기 구성 순차 탐색(Self-Organizing Sequential Search)

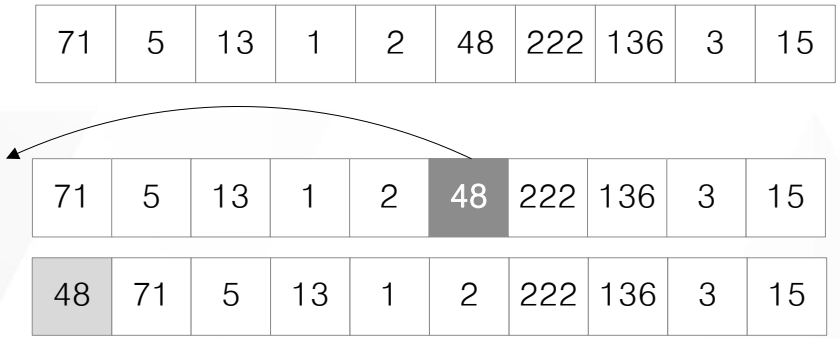
- 자주 사용되는 항목을 데이터 집합의 앞쪽에 배치함으로써 순차 탐색의 검색 효율을 끌어올리는 방법

  - 전진 이동법(Move To Front)
  - 전위법(Transpose)
  - 빈도 계수법(Frequency Count)

3 자기 구성 순차 탐색(Self-Organizing Sequential Search)

전진 이동법(Move to front)

- 어느 항목이 한번 탐색되고 나면
  그 항목을 데이터 집합의 가장 앞에 위치시키는 방법



3 자기 구성 순차 탐색(Self-Organizing Sequential Search)

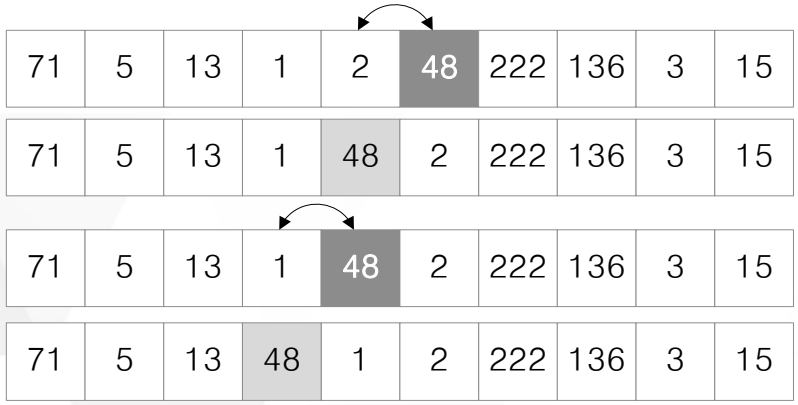
전위법(Transpose)

- 탐색된 항목을 바로 이전 항목과 교환함

3 자기 구성 순차 탐색(Self-Organizing Sequential Search)

전위법(Transpose)

- 전진 이동법과는 달리 자주 사용되면 사용될수록 점진적으로 해당 항목을 데이터 집합의 앞쪽으로 이동시킴



3 자기 구성 순차 탐색(Self-Organizing Sequential Search)

빈도 계수법(Frequency count)

- 데이터 집합내의 각 요소들이 탐색된 횟수를 별도의 공간에 저장해 두고 탐색된 횟수가 높은 순으로 데이터 집합을 재구성
- 계수 결과를 저장하는 별도의 공간을 유지해야 하고 계수 결과에 따라 데이터 집합을 재배치해야 함

3 이진 탐색

3 이진 탐색

1 이진 탐색(Binary Search)

- 정렬된 데이터 집합을 이분화하면서 탐색하는 방법
- 정렬되어 있는 전체 파일을 두 개의 서브파일로 분리해가면서 키값을 검색
  - 예 : 가나다순으로 정렬되어 있는 전화번호부에서 임의의 사람에 대한 전화번호를 찾는 경우

순차 탐색은 데이터가 정렬되어 있을 때도 데이터를 하나씩 비교하면서 원하는 값을 찾기 때문에 비효율적

3 이진 탐색

1 이진 탐색(Binary Search)

- 찾고자 하는 키값을 파일의 중간 레코드 키값과 비교하면서 검색
- 시작과 끝의 인덱스를 설정하고 그 중간값을 구한 후 키와 중간값을 계속 비교하면서 검색
- 파일의 탐색 시간을 1/2씩 줄여가면서 탐색하므로 효율적
- 레코드의 수가 많을수록 효과적

3 이진 탐색

1 이진 탐색(Binary Search)

- 중간 레코드 번호

$$m = \frac{l + u}{2}$$

(단, l : 첫 번째 레코드 번호, u : 마지막 레코드 번호)

- 시간 복잡도: O(logn)

3 이진 탐색

1 이진 탐색(Binary Search)

◆ **정렬된 데이터 집합**에서 사용할 수 있는 '고속' 탐색 알고리즘

① 데이터 집합의 중간에 있는 값 선택

↓

② 중간값과 찾고자 하는 목표값을 비교

↓

③ 목표값이 중간값보다 작다면 중간값의 왼편에 대해
검색을 수행하고 크다면 오른편에 대해 이진 탐색을 수행

↓

④ 찾고자 하는 값을 찾을 때까지 ①번~③번 과정을 반복

3 이진 탐색

2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

[이진 탐색을 위한 데이터 집합]

[0]	[1]	[2]	[3]	[4]	[5]	[6]
1	3	8	11	15	17	20

중간값(m)= $\frac{l+u}{2} = \frac{0+6}{2} = 3$

3 이진 탐색

2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

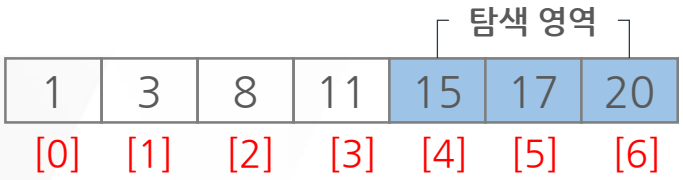
① 배열의 인덱스가 3인 위치에 있는 데이터인 11과 찾고자 하는 15가 같은지 비교

3 이진 탐색

2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

② 중간에 위치한 데이터인 11보다 찾고자 하는 데이터인 15가 크므로 중간 데이터 11의 오른쪽에 위치한 데이터들에 대해 이진 탐색을 수행

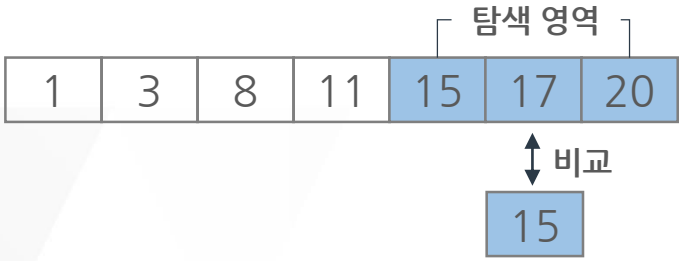


중간값(m)= $\frac{l+u}{2} = \frac{4+6}{2} = 5$

2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

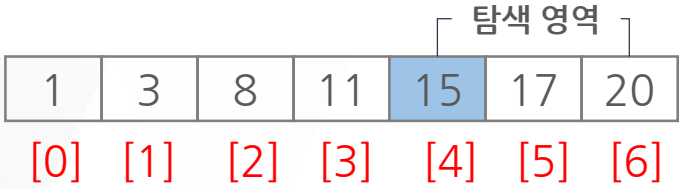
③ 배열의 인덱스가 5인 위치에 있는 데이터인 17과 찾고자 하는 15가 같은지 비교



2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

④ 중간에 위치한 데이터인 17보다 찾고자 하는 데이터인 15가 작으므로 중간 데이터 17 왼쪽에 위치한 데이터들에 대해 이진 탐색 수행



중간값(m)= $\frac{l+u}{2} = \frac{4+4}{2} = 4$

3 이진 탐색

2 이진 탐색의 작동 예

◆ 예) 다음의 정렬된 데이터 집합에서 이진 탐색을 이용해 데이터 15를 탐색하는 과정

⑤ 배열의 인덱스가 4인 위치에 있는 데이터인 15와 찾고자 하는 15가 같은지 비교하고 같으므로 탐색 종료