



# 컴퓨팅사고력 향상을 위한 문제해결과 알고리즘

1교시 : 자료탐색 알고리즘 1

양 속 희

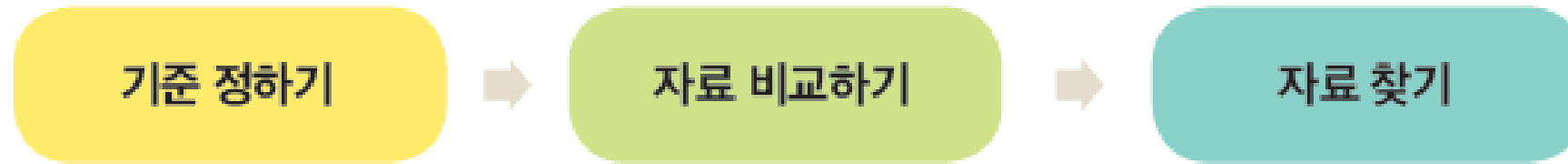
# 자료 탐색 알고리즘

1. 탐색 알고리즘의 개념을 이해 할 수 있다.
2. 탐색 알고리즘의 종류별 특징을 이해 할 수 있다.
3. 탐색 알고리즘을 선택하여 문제를 해결 할 수 있다

- 수많은 자료 중에서 원하는 자료를 찾는 작업
- 컴퓨터에서 처리되는 핵심 알고리즘 중 하나에 해당
- **탐색키(search key)**
  - 탐색할 때 기준이 되는 값
  - 항목과 항목을 구별하는 값

- 탐색 키에 해당하는 인덱스 번호 반환
  - 인덱스 번호란
    - 자료 리스트 목록의 순차적 번호를 의미
    - 첫 번째 번호는 '0'부터 시작
- 예: [2, 4, 7, 8, 12]에서 7을 탐색하는 경우
  - 인덱스 번호 2가 반환
- 탐색하고자 하는 자료가 리스트 안에 포함되어 있지 않는 경우
  - 일반적으로 -1을 반환

자료 탐색의 과정은 다음의 3단계를 거쳐서 이루어짐



## 1단계 : 기준 정하기

- 탐색의 기준이 되는 탐색키를 선정하는 작업

## 2단계 : 자료 비교하기

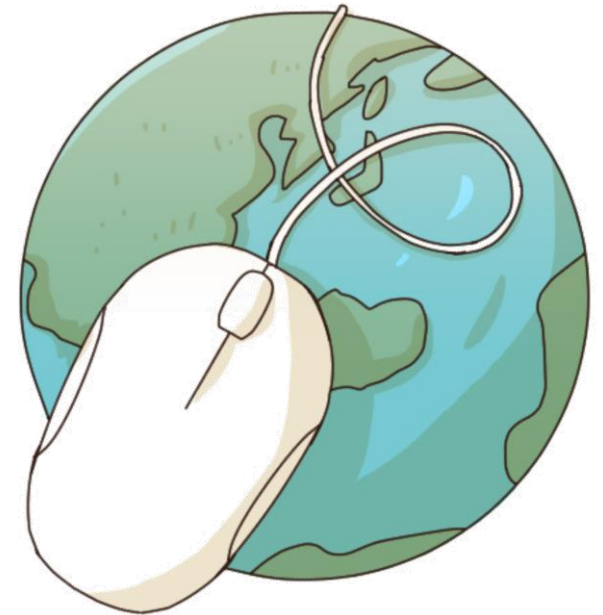
- 탐색키와 자료 리스트에 포함된 자료들의 실질적 비교작업 진행
- 자료 비교 방법에 따라 자료 탐색 알고리즘 효율성의 차이가 발생

## 3단계 : 자료 찾기

- 탐색의 성공과 실패에 대한 판가름

## 인터넷에서 자료를 찾을 때

- 탐색키 : 해당하는 검색어
- 검색어와 탐색 방법에 따라 탐색 결과에 차이가 발생



① 선형탐색 알고리즘 (Linear Search Algorithm) = Sequential Search

② 이진 탐색 알고리즘(Binary Search Algorithm)

③ 해시 알고리즘(Hash Algorithm)

파이썬의 딕셔너리 : 해시테이블(hashtable)이라고 불리는 알고리즘을 사용



- **리스트** : 리스트는 **위치(index)**에 의하여 관리되는 자료구조
- **딕셔너리**: 오직 **탐색 키(key)**에 의해서만 관리되는 자료구조

**사전 구조(dictionary) : 해시, 연관배열**

사전 구조는 **탐색 키(key)**, 그리고 **탐색 키와 관련 있는 값(value)**

- 영어 단어나 사람의 이름 같은 **탐색 키 (saerch key)**
- 단어의 정의나 주소 또는 전화 번호와 같은 **탐색 키와 관련된 값 (value)**

## Sequential Search

- 문제 정의
  - 처음부터 끝까지 하나하나 차례대로 비교하며 찾고자 하는 값을 탐색하는 방식
- 자료가 정렬되지 않은 상태에서 특정 자료의 값을 찾는 경우에 적절
- 자료가 존재한다면 자료의 위치 번호(index 값)를 반환
  - 자료가 존재하지 않는다면 탐색에 실패한 경우로, 일반적으로 -1을 반환

## 【예】 주어진 자료에서 46 탐색

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

index 0

3과 46은 같지 않음

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

생략 ...

index 10

46과 46은 같음 : 탐색 성공

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	5	10	12	14	16	19	30	42	45	46	47	53	55	62	69	70

# 메뉴 함수정의

```
def menu():  
    print()  
    print('Friends and telephone')  
  
    print('='*30)  
    print('1. Look up a telephone')  
    print('2. Add a new telephone')  
    print('3. Delete a telephone')  
    print('4. All telephone')  
    print('5. Quit the program')  
    print('='*30)
```

# 딕셔너리 활용한 주소록 만들기 - 메인코드1

```
# 메인코드
menu()
addr={} #

while True:
    choice = int(input('Enter your choice: '))

    if choice==1: #
        name=input('Enter a name: ')
        print( ) #
```

# 딕셔너리 활용한 주소록 만들기 - 메인코드2

```
while True:
    choice = int(input('Enter your choice: '))

    if choice==2: #
        name=input("Enter a name:")
        if         :
            print("already exists")
        else:
            tel=input("Enter a telephone:")
                        
```

# 딕셔너리 활용한 주소록 만들기 - 메인코드3

```
while True:
    choice = int(input('Enter your choice: '))

    if choice==3: #
        name=input("Enter a name:")
        if name in addr:
            #
        else:
            print("not Found")
```

```
# 함수정의
def search_list(key, data):
    n=len(data)
    for i in range(n):

        if [redacted] : #

            return i

    return -1
```

```
# 메인코드
data=[8,6,5,2,4]
print(data)
key=int(input('search key : ')) #

[redacted] #

if index==-1:
    print('탐색 실패')
else:
    print('탐색 성공! 탐색 위치= %d' % index)
```



# 자료 탐색 알고리즘

자료 탐색이란

자료 탐색 과정

- 기준 정하기
- 자료 비교하기
- 자료 찾기

자료 탐색 알고리즘 기법

- 리스트 형식의 자료



수고하셨습니다.

