

파이썬으로 배우는 알고리즘 기초

Chap 1. 알고리즘: 효율, 분석, 차수



1.1

알고리즘이란?





1.1 알고리즘이란?



■ 알고리즘의 정의

- 어떤 문제를 컴퓨터로 풀기 위한 효율적인 절차
- 문제를 푸는 단계별 절차를 명확하게 기술

■ 알고리즘을 공부하는 목적

- 어떤 문제를 컴퓨터로 해결하는 방법을 공부함
- 특정 프로그래밍 언어나 문법과는 무관함
- 다양한 문제 해결 방법(=알고리즘 설계 기법)을 공부함
- 알고리즘 문제를 이해하고 효율적으로 해결하는 방법을 공부함
- 새로운 문제를 만났을 때, 그것을 해결할 수 있는 능력을 배양함



1.1 알고리즘이란?



■ 문제와 해답

- 문제란 해답을 찾으려고 물어보는 질문
- **파라미터**는 문제에서 특정한 값이 지정되어 있지 않은 변수
- **입력 사례**란 문제의 파라미터에 지정된 특정한 값
- 특정 입력사례의 해답은 해당 파라미터를 입력사례로 질문한 문제의 해답

■ 알고리즘

- 어떤 문제의 **모든 입력 사례**에 대해서 해답을 찾아주는 단계별 절차
- 입력 파라미터에 어떤 입력 사례가 주어지더라도 해답을 찾을 수 있어야 함



1.1 알고리즘이란?



■ 순차 탐색 문제

- 문제: 어떤 수 x 가 n 개의 수로 구성된 리스트 S 에 존재하는가?
- 해답: x 가 존재하면 x 의 인덱스가, 존재하지 않으면 0이 해답
- 파라미터: 정수 $n(> 0)$, 리스트 S (인덱스 범위는 1부터 n 까지), 원소 x
- 입력 사례: $S = [0, 10, 7, 11, 5, 13, 8], n = 6, x = 5$
- 입력 사례에 대한 해답: $location = 4$
- 알고리즘: 모든 S 에 대해서 x 의 인덱스를 찾아주는 단계별 절차
 - S 의 첫째 원소에서 시작하여 x 를 찾을 때까지 (x 가 없는 경우 끝까지)
 - 각 원소를 차례로 x 와 비교한다.
 - 만약, x 를 찾으면 x 의 인덱스를 리턴하고,
 - x 를 찾지 못하면 0을 리턴한다.



1.1 알고리즘이란?



Algorithm 1.1: Sequential Search

```
def seqsearch (n, S, x):  
    location = 1  
    while (location <= n and S[location] != x):  
        location += 1  
    if (location > n):  
        location = 0  
    return location
```



1.1 알고리즘이란?



```
S = [0, 10, 7, 11, 5, 13, 8]
```

```
x = 5
```

```
location = seqsearch(len(S) - 1, S, x)
```

```
print('location =', location)
```

```
x = 4
```

```
location = seqsearch(len(S) - 1, S, x)
```

```
print('location =', location)
```



1.1 알고리즘이란?



■ 리스트(배열) 원소의 합 구하기

- 문제: n 개의 원소를 가진 리스트(배열) S 의 원소의 합을 구하시오.
- 해답: 리스트(배열) S 의 모든 원소들의 합
- 파라미터: 리스트 S , 정수 n
- 입력 사례: $S = [-1, 10, 7, 11, 5, 13, 8]$, $n = 6$
- 출력 사례: $sum = 54$
- 알고리즘: S 의 모든 원소를 차례대로 sum 에 더하는 절차
 - sum 을 0으로 초기화
 - 모든 S 의 원소에 대해서 $sum += S[i]$ 를 실행
 - sum 의 값을 리턴



1.1 알고리즘이란?



Algorithm 1.2: Add Array Elements

```
def sum (n, S):  
    result = 0  
    for i in range(1, n + 1):  
        result = result + S[i]  
    return result
```

```
S = [-1, 10, 7, 11, 5, 13, 8]  
sum = sum(len(S) - 1, S)  
print('sum =', sum)
```




1.1 알고리즘이란?



■ 리스트의 정렬 문제

- 문제: n 개의 수로 구성된 리스트 S 를 비내림차순으로 정렬하시오.
- 해답: S 를 비내림차순으로 정렬한 리스트
- 파라미터: S, n
- 입력 사례: $S = [-1, 10, 7, 11, 5, 13, 8]$
- 입력 사례에 대한 해답: $S' = [-1, 5, 7, 8, 10, 11, 13]$
- 알고리즘: 모든 S 에 대해서 S' 을 찾아주는 단계별 절차
 - 교환 정렬, 삽입 정렬, 선택 정렬, 합병 정렬, 퀵 정렬, 기타 등등.
 - 여러 가지 정렬 알고리즘 중에서 교환 정렬 방법으로 구현.



1.1 알고리즘이란?

Algorithm 1.3: Exchange Sort

```
def exchange (S):  
    n = len(S)  
    for i in range(n - 1):  
        for j in range(i + 1, n):  
            if (S[i] > S[j]):  
                S[i], S[j] = S[j], S[i] # swap
```



1.1 알고리즘이란?



```
S = [-1, 10, 7, 11, 5, 13, 8]
print('Before =', S)
exchange(len(S) - 1, S)
print('After =', S)
```

```
Before = [-1, 10, 7, 11, 5, 13, 8]
After = [-1, 5, 7, 8, 10, 11, 13]
```



1.1 알고리즘이란?



■ 교환 정렬

- i 번째 자리에 있는 수와 $(i + 1)$ 번째부터 n 번째 자리에 있는 수를 차례대로 비교
- 주어진 자리의 수가 i 번째 자리에 있는 수보다 작은 경우, 두 수를 교환
- for- i 루프를 한 번 수행하면 그 중 가장 작은 수가 첫번째 자리에 들어감
- 두번째 루프를 수행하면 둘째 자리에 둘째로 작은 수가 들어감
- 이런 과정을 통해 for- i 루프가 모두 수행되면 비내림차순 정렬이 됨



1.1 알고리즘이란?



■ 행렬의 곱셈 문제

- 두 $n \times n$ 행렬의 곱을 구하시오

- 입력 사례: $n = 2$

- $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$

- $\begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 28 & 38 \\ 26 & 36 \end{bmatrix}$

- $C = A \times B, c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j}$

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

for all $1 \leq i, j \leq n.$



1.1 알고리즘이란?



Algorithm 1.4: Matrix Multiplication

```
def matrixmult (A, B):  
    n = len(A)  
    C = [[0] * n for _ in range(n)]  
    for i in range(n):  
        for j in range(n):  
            for k in range(n):  
                C[i][j] += A[i][k] * B[k][j]  
    return C
```



1.1 알고리즘이란?



```
A = [[2, 3], [4, 1]]
```

```
B = [[5, 7], [6, 8]]
```

```
print('A =', A)
```

```
print('B =', B)
```

```
C = matrixmult(A, B)
```

```
print('C =', C)
```

```
A = [[2, 3], [4, 1]]
```

```
B = [[5, 7], [6, 8]]
```

```
C = [[28, 38], [26, 36]]
```



주니온TV@Youtube

자세히 보면 유익한 코딩 채널

<https://bit.ly/2JXXGqz>

주니온TV@Youtube

자세히 보면 유익한 코딩 채널

- 여러분의 **구독**과 **좋아요**는 강의제작에 큰 힘이 됩니다.
- 강의자료 및 소스코드: **구글 드라이브**에서 다운로드
(다운로드 주소는 영상 하단 설명란 참고)

<https://bit.ly/3fN0q8t>