


2. 알고리즘 시간 복잡도

🕒 작성일	@2021년 8월 30일 오후 12:37
👤 작성자	 김현빈

자료구조, 알고리즘 시간 복잡도(time complexity)

예시:

```
A = [3, -1, 9, 2, 12]
n = 5

def algorithm ArrayMax(A, n):
    # input: n개의 정수를 갖는 배열 A
    # output: A의 수 중에서 최대값을 리턴

    currentMax = A[0]

    for i in range(1,n-1): # 단위시간 1
        if currentMax < A[i]: # 단위시간 1
            currentMax = A[i] # 단위시간 1
    return currentMax

# 반복된 각각의 기본연산 횟수의 합을 수행시간이라 하고,
# 여기서는 7번이 반복되기 때문에 7단위 시간이라고 한다.
```

위와 같은 예시는 모든 입력에 대해 기본 연산 횟수를 더한 후 평균을 내어 시간 복잡도를 정하는 방법이다. 하지만 이는 고려해야 할 점이 너무 많기 때문에 현실적으로 불가능하다.

따라서 우리는 **알고리즘 수행 시간**을 기본 연산 횟수를 가장 많이 요구하는 **'가장 안 좋은 입력(Worst Case Input)'**에 대한 기본 연산 횟수를 측정한 시간이라고 한다. 이는 정확성이 떨어지지만, **어떤 입력이라도 최악의 입력을 대입한 WTC보다 수행 시간이 크지 않다는 것이 보장되는 것이 가장 큰 장점이다.** 이를 기반으로 위의 예제를 다시 보자.

1. $currentMax = A[0]$ 이 1번 수행되고
2. for문이 $1 \sim n(-1)$ 번 수행되며
3. if $currentMax < A[1]$ 이 참일 때 실행되는 연산이 가장 길기 때문에 for문 안은 2번씩 수행된다.

따라서 WTC의 알고리즘 수행 시간 $T(n) = 1 + (n-1)*2 = 2n-1$ 이 된다. $n=5$ 이므로 $T(5) = 9$ 가 된다.

연습 문제 1:

```
def algorithm sum1(A, n):
    # input: n개의 정수를 갖는 배열 A
    # output: A의 수 중에서 최대값을 리턴

    sum = 0

    for i in range(0, n-1): # 단위시간 1
        if A[i] % 2 == 0: # 조건: 짝수일때 수행, 단위시간: 2(% ==)
            sum += A[i] # 단위시간 2(+, =)
    return sum
```

정답:

A: 모든 값이 짝수일 경우 Worst Case.

$$T(n) = 1 + n * 2 * 2 = 4n + 1$$

연습 문제 2:

```
def algorithm sum2(A, n):
    sum = 0
    for i in range(0, n-1):
        for j in range(i, n-1):
            sum += A[i] * A[j]

    return sum
```

정답:

$$T(n) = 1 + (1+2+3+4+5+6+...+n) * 3 = 1 + \{n(n+1)/2\} * 3 = 3/2 * n * (n+1) + 1 = 3/2 n^2 + 3/2 n + 1$$