


1. 자료구조와 알고리즘 성능

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

자료구조와 알고리즘을 수행하는 실제 컴퓨터는 개인마다 하드웨어와 소프트웨어 환경이 서로 상이하다. 따라서 동일한 언어(C, Java, Python)로 동일한 알고리즘을 수행하였다고 하여도, 개인마다 서로 다른 성능을 보여주게 된다. 이는 매우 큰 문제가 될 수 있기 때문에 우리는 여러가지 종류의 다양한 크기의 입력의 수행에 대해, 보다 객관적으로 측정할 수 있는 방법을 생각해봐야 한다.

객관적인 가상 컴퓨터

따라서 우리는 하드웨어와 소프트웨어 환경에 구애 받지 않는, 독립적이고 객관적인 가상 컴퓨터(Virtual Machine)를 정의한다. 이 가상 컴퓨터에서 사용되는 언어를 가상 언어(Pseudo Language)라고 하고, 그에 대한 코드를 가상 코드(Pseudo Code)라고 한다.

가상 컴퓨터:

과거의 Turing Machine이 발전하여 현재 컴퓨터와 매우 유사한 모델이 제시되었다. 이는 RAM으로 Random Access Machine이라고 불린다. 이 컴퓨터 모델(RAM)은 CPU + Memory + 기본 연산으로 이루어진다. 메모리에는 프로그램이 다루는 모든 데이터가 저장되고, CPU가 메모리에 접근해서 데이터를 읽고 쓰고 수정하는 가공 역할을 한다. 기본 연산은 단위 시간에 수행되는 연산들의 모음이다. 이는 가장 간단한 형태의 연산으로,

• 연산 종류

1. 배정 연산, 대입 연산, 복사 연산 ex) $A=B$ - 1단위 시간 내에 작동

2. 산술 연산: ex) $+$, $-$, $*$, $/$ - 1단위 시간 내에 작동.

cf) 하지만 나머지 연산($\%$) 버림 연산, 올림, 반올림 연산 등은 단위 시간 내에 수행되는 연산이라고

볼 수 있지만, RAM 모델에서는 기본 연산이라고 정의 되지 않는다.

3. 비교 연산: ex) $>$, \geq , $<$, \leq , $=$, \neq - 1단위 시간 내에 작동. 이는 A와 B를 -하여 0보다 큰지 판별하는 것과 같다.

4. 논리 연산: ex) AND, OR, NOT

5. 비트 연산: ex) bit-AND, OR, NOT

가 있다. 즉 알고리즘은 RAM이라는 가상 머신(Virtual Machine) 위에서 돌아가게 된다. 그리고 알고리즘은 기본 연산을 반복적으로 수행하며, CPU와 Memory와 함께 작동하여 특정한 일을 하는 것이다.

가상 언어(Pseudo/Virtual Languages)

가상 머신(컴퓨터)위에 동작하는 언어는 RAM모델에서 제공하는 기본 연산들을 표현할 수 있는 언어이면 되는 것이고, 여러 제어 명령어의 집합을 제공해줄 수 있으면 되는 것이다. 이는 가상 컴퓨터에서 명령어에 따라서 시뮬레이션되는 것이다. 이는 우리가 사용하는 말보다는 명확하고, 프로그램 언어보다는 느슨한 언어면 된다.

- 필수 조건
 - 배정, 산술, 비교, 논리, bit-논리 등 기본 연산을 표현할 수 있으면 된다.
 - 비교: if, else, elseif,
 - 반복: for, while
 - 함수: 정의, 호출, return이 가능하면 된다.

이런 필수 조건을 다 제공한다면 모두 가상의 언어라고 한다.

가상 코드(Pseudo Code)

이 언어를 사용하여 가상 머신에서 돌아가는 코드를 가상 코드(Pseudo Code)라고 한다.

EX]

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

def algorithm Array Max(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.
```

