

Introduction to Python and Programming Language

Koscom Algorithm Lecture

신승우

Tuesday 4th September, 2018

- 1 Computation and Computation Model
- 2 Concept of Computation
 - Computation Model
 - Computability
 - Accepting Subroutines : Concept of Oracle
- 3 ADT vs Data Structure
 - Abstraction
 - Data Structure
- 4 Recursion
 - Analyzing Recursion
 - Dynamic Programming

Motivating Example

소인수분해 알고리즘

- 일반 컴퓨터 : $O(n^k)$
- 양자 컴퓨터 : $O(\log^k n)$

왜 이런 차이가 날까요?

Computation이란?

- **Initial Setting**에서 시작해서
- **정해진 operation**들을 유한 번 거쳐
- Output을 결정

하는 과정을 Computation이라고 한다. 여기서 Initial Setting과 정해진 operation들은 어떻게 정해질까요? 이것은 Computational Model에 의해서 정해집니다.

Computation Model은 정의하기에 따라 매우 다양한데, 보통 많이 쓰이는 모델들은 다음과 같습니다.

- RAM Model
- Turing Machine
- Decision Trees

우리가 일반적으로 생각하는 프로그래밍 언어들은 대부분 Turing Machine과 같은 계산력을 가집니다. 즉, 튜링머신으로 풀 수 있는 문제는 일반 프로그래밍 언어로 풀 수 있고, 그 반대도 성립합니다.

Halting Problem

파이썬 언어로 되어있는 어떤 임의의 함수 f 에 대해서, 이 함수가 멈추는지 알려주는 함수를 파이썬으로 짤 수 있을까?

Concept of Oracle

Computation Model은 종종 오라클이라는 외부 계산기기를 가지기도 합니다. 이 때, computation model은 기존의 operation에 더해서, 이 오라클에 input을 넣고 output을 받는 operation을 할 수 있습니다. 이 오라클 내에서의 계산과정은 black box이며, 원 computation model에서는 전혀 알 수 없습니다.

이는 프로그래밍에서 subroutine의 개념과 정확히 일치합니다. 이러한 subroutine 중 많이 쓰이는 것들을 모아놓은 것이 Abstract Data Type 입니다.

Motivating Example

순서에 대해서 생각해 보자.

- Total Order : 정수 간의 순서 / 문자열 간의 순서 / ...
- Partial Order : 대학교 과목 간의 순서 / 할 일의 순서 / ...

개별적인 문제에서, 공통된 부분을 추출하는 것!

많이 쓰이는 **operation**들을 모아놓은 것. 예를 들면,

- 선입선출되는 상황에서 많이 쓰이는 operation set : Queue
- 선입후출되는 상황에서 많이 쓰이는 operation set : Stack
- 계층구조 등을 나타낼 때 많이 쓰이는 operation set : Tree

로 생각할 수 있습니다. 여기서, operation들만 모아놓은 것이지, 어떤 식으로 그 operation을 구현할지는 전혀 언급되지 않았습니다.

Interface vs Implementation

위와 같은 operation set들은 함수들 간의 일종의 인터페이스를 제공합니다. 위 오라클 머신에서, 오라클의 계산 과정은 전혀 밝혀지지 않았음을 상기하세요. 우리가 아는 것은 함수들의 집합일 뿐입니다. 자바에서는 이와 같은 개념을 interface라 하며, C++에서는 Abstract class라고 합니다. 파이썬에서는 abc 모듈을 이용하여 나타낼 것입니다. 현실에서 이 인터페이스를 쓰기 위해서는 이를 구현해야 합니다. 이를 어떻게 구현할지 모아 놓은 것을 데이터구조라고 합니다.

데이터구조의 예시에는 다음과 같은 것들이 있습니다.

- Array
- Linked List : Doubly Linked List, Cyclic Linked List, ...
- Binary Tree
- Heap : Binary Heap, Fibonacci Heap, ...

Example : Priority Queue

Priority Queue를 구현하기 위해서, 다음의 두 데이터구조를 생각해 봅시다.

- Linked List
- Sorted Linked List : 새 원소가 들어올 때, 리스트가 언제나 정렬된 상태를 유지하도록 함.

이 때, Priority Queue의 operation들의 복잡도는 어떻게 될까요?

Recursive Function Example : Fibonacci Sequence

실습 : 피보나치 수열은 $f(n) = f(n-1) + f(n-2)$ 를 만족하는 수열을 말합니다. 이 때, 피보나치 수를 계산하는 함수를 짜 보면 어떻게 될까요?

Computational Complexity : Fibonacci Sequence

이 때, 위에서 짚 알고리즘의 복잡도는 어떻게 계산할 수 있을까요?

Fibonacci Sequence Revisited

실습 : 위에서 짰 알고리즘이 복잡한 이유는 같은 것을 여러 번 계산하기
때문입니다. 따라서, 이를 개선하기 위해서 기존에 계산한 것을 저장하는
리스트를 만들고, 그 리스트를 업데이트하면서 계산해 봅시다.

Fibonacci Sequence Revisited

추가 실습 : 더 빠른 방법은 없을까요? 행렬을 사용해 보세요.