

POINTER 알고리즘 세미나

제 2장 - 자료 구조 (Data structure)

Kim Jun Hyeok

Kangwon Science Highschool

September 5, 2020

1 개요

1.1 자료 구조의 필요성

다음과 같은 3가지의 연산을 빠르게 하기 위해서 등장함.

- 삽입 (Insertion)
- 삭제 (Deletion)
- 탐색 (Search)

1.2 자료 구조의 분류

선형 자료구조 데이터가 순차적으로 연결되어있는 방법으로 표현된 자료구조.
ex) 배열, 연결 리스트, 스택, 큐등

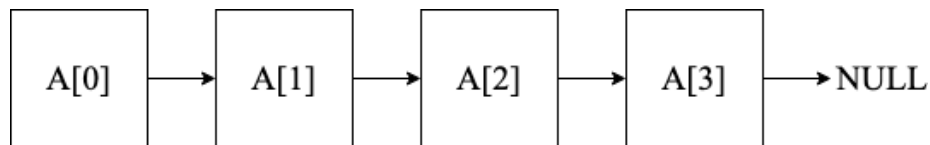


Figure 1: 선형 자료구조의 예시: 연결 리스트 (Linked list)

비선형 자료구조 데이터가 순차적으로 연결되어있지 않은 방법으로 표현된 자료 구조. ex) 그래프, 트리등

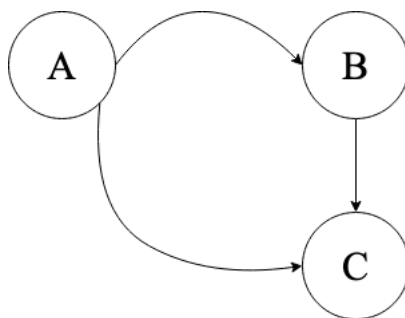


Figure 2: 비선형 자료구조의 예시: 그래프(Graph)

2 선형 자료 구조

2.1 스택(Stack)

개요 스택(Stack) 자료 구조는 먼저 집어넣은 원소가 가장 먼저 나오게 하는 LIFO(Last-In, First-Out) 형식을 따르는 자료 구조이다. [1] 이러한 스택(Stack) 자료 구조는 동적 할당 배열(Dynamic array)로 구현할 수 있다.

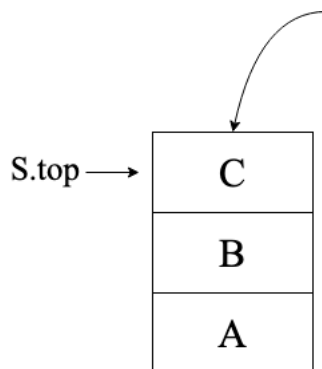


Figure 3: 스택(Stack) 자료 구조를 표현한 그림

속성

- top - 가장 최상위의 원소를 반환함.

기본 연산 스택(Stack) 자료 구조에서 기본 연산은 다음과 같다.

- $PUSH(s)$ - top 에 원소를 삽입함.
- $POP()$ - top 에 있는 원소를 제거함.
- $STACK - EMPTY()$ - 스택이 비었는지 확인함.

구현 C++ 11 표준안을 기준으로 `std::stack`을 통해서 사용할 수 있음.

2.2 큐(Queue)

개요 큐(Queue) 자료 구조는 먼저 집어넣은 원소가 나중에 나오게 하는 FIFO(First-In, First-Out) 형식을 따르는 자료 구조이다. [1]

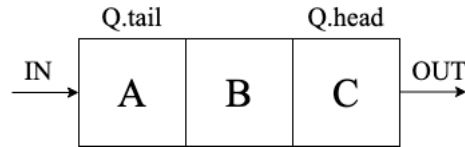


Figure 4: 큐(Queue) 자료 구조를 표현한 그림

속성

- $tail$ - 가장 처음에 삽입된 원소를 반환함.
- $head$ - 가장 나중에 삽입된 원소를 반환함.

기본 연산 큐(Queue) 자료 구조에서 기본 연산은 다음과 같다.

- $ENQUEUE(s)$ - $tail$ 에 원소를 삽입함.
- $DEQUEUE()$ - $head$ 에 있는 원소를 제거함.

References

- [1] T. H. e. a. Cormen, *Introduction to algorithms*. MIT Press, 2009.