



D.COM ALGORITHM STUDY

2020 - 01 VER.

2강) 자료구조!

2강. 자료구조!

오늘의 목표

첫번째로, 자료구조는 왜 배워야 할까요?

두번째로, 자료구조의 종류와
스택, 큐, 덱에 대해서 알아보시다.

#STL 사용법도 알려드립니다!

가볍게,
읽을거리



링크

프로그래머로서의 성장을 도왔던 태도들

“세상은 넓고 나는 작지만
내가 할 수 있는 일이 있다”



자료구조를 **왜** 배워야 할까요?

~~아니 자료구조 학점 따려고 배우겠지~~

자료구조를 통해 복잡한 문제를
간단하게

추상화 및 **단순화**

할 수 있습니다.

Q.

여러분은

지하철 길 찾기

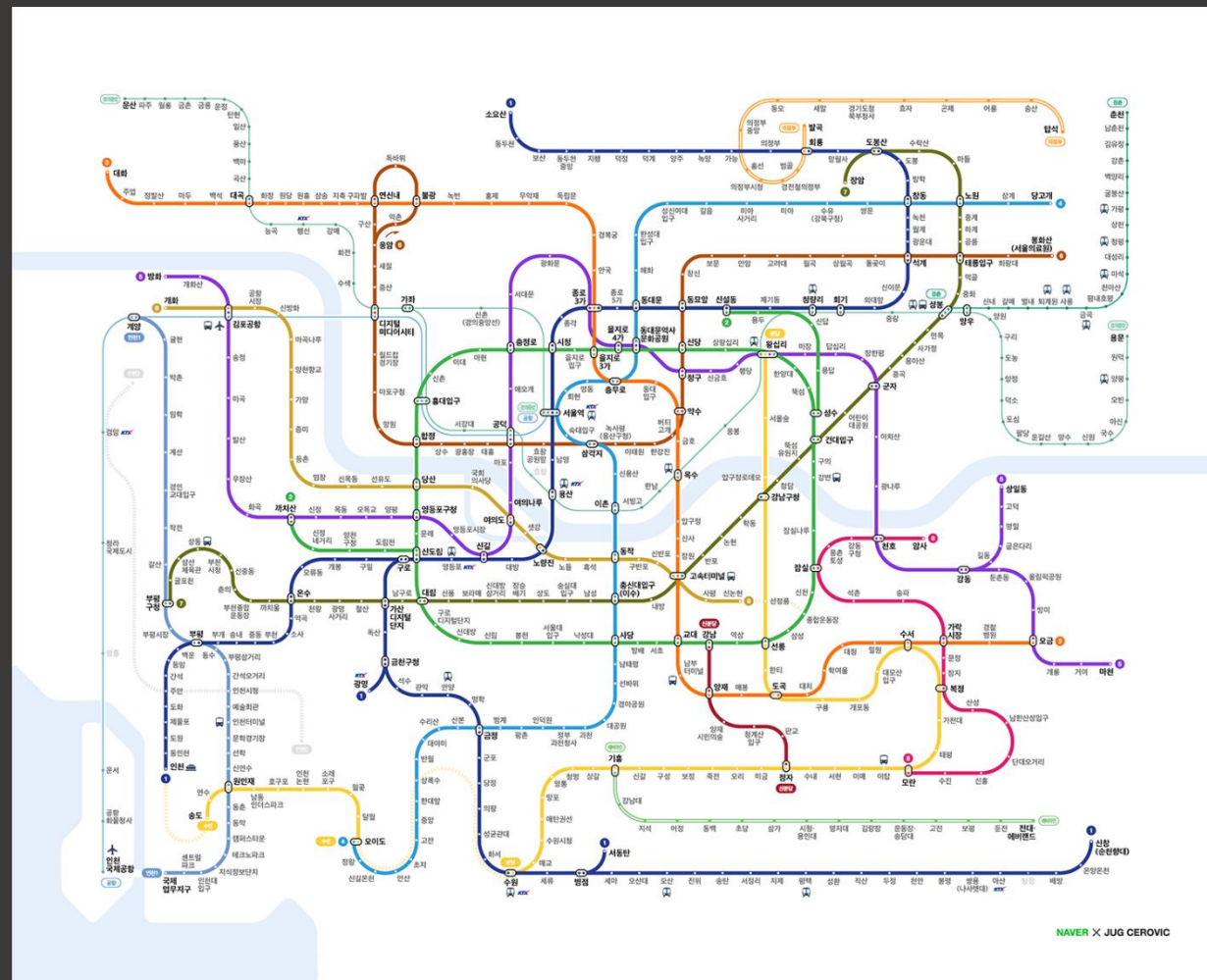
프로그램을 제작하고 싶습니다.
어떻게 해야 할까요?



실제 지하철 역과 노선의
좌표 데이터를 이용하면
문제가 매우 **복잡**해 집니다.

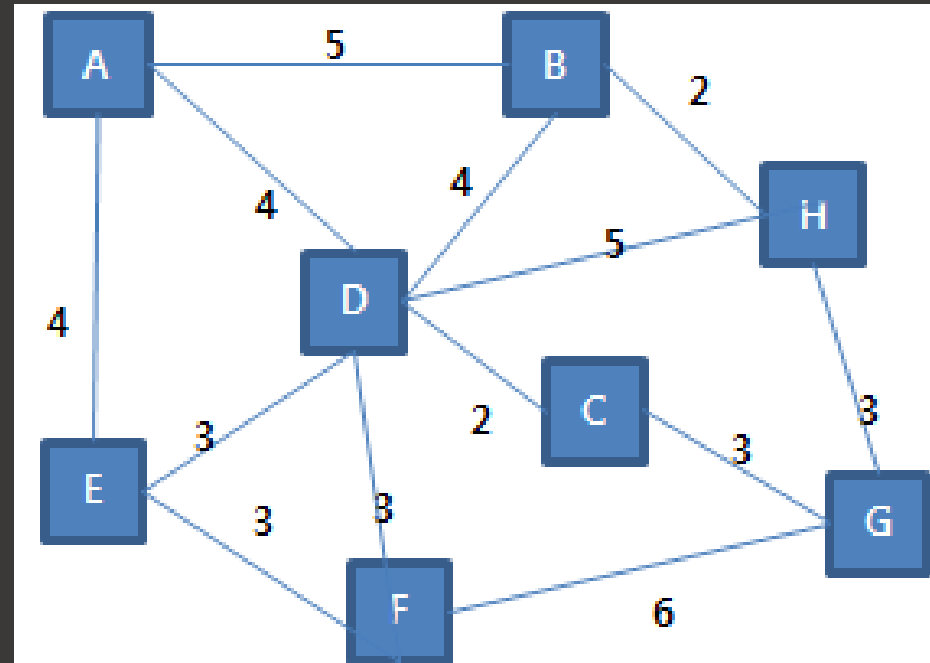


따라서 우리는 지하철 노선을
다음과 같이
추상화하고 **단순화**하여
사용합니다.



더 나아가,
실제로 **그래프**라는 자료구조를 이용하여
지하철 역간 최적의 경로를 구할 수 있습니다.

#그래프는 추후 강의에서 자세히 다룹니다.



이처럼,
자료구조를 통해서
문제를
추상화하고 단순화하여
문제 해결의 실마리를 찾을 수 있습니다.

또한,

적절한 자료구조를 사용하면

자료를 더욱 **효율적**으로 저장 및 관리 할 수 있습니다.

이는 실행 시간 단축과 메모리 용량의 절약을
이끌어 낼 수 있습니다.



그래서,

자료구조란 무엇일까요?

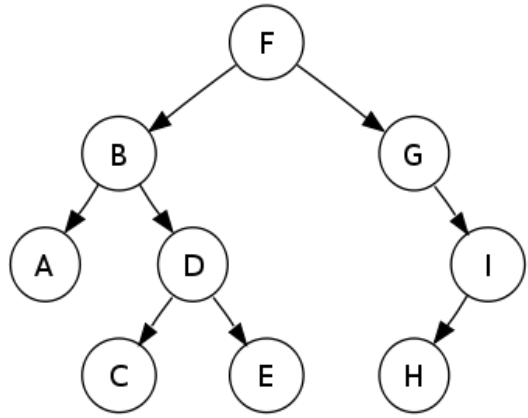


자료구조

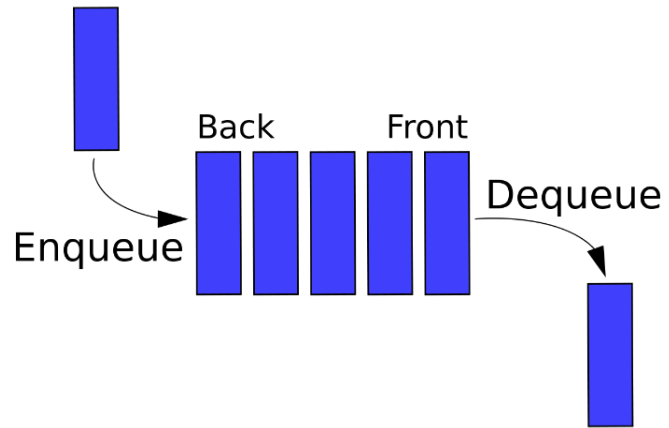
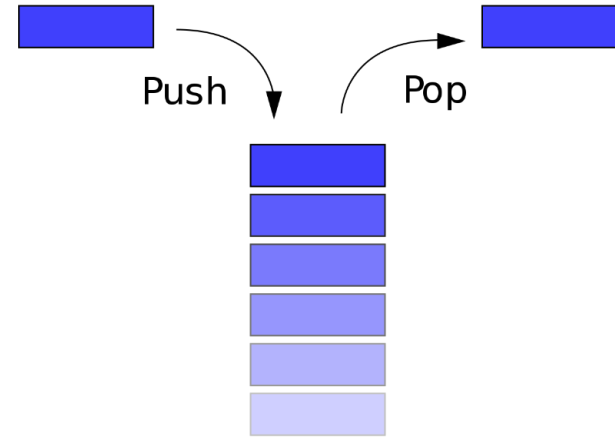
Data Structure

데이터를 표현하고 저장하는 방법

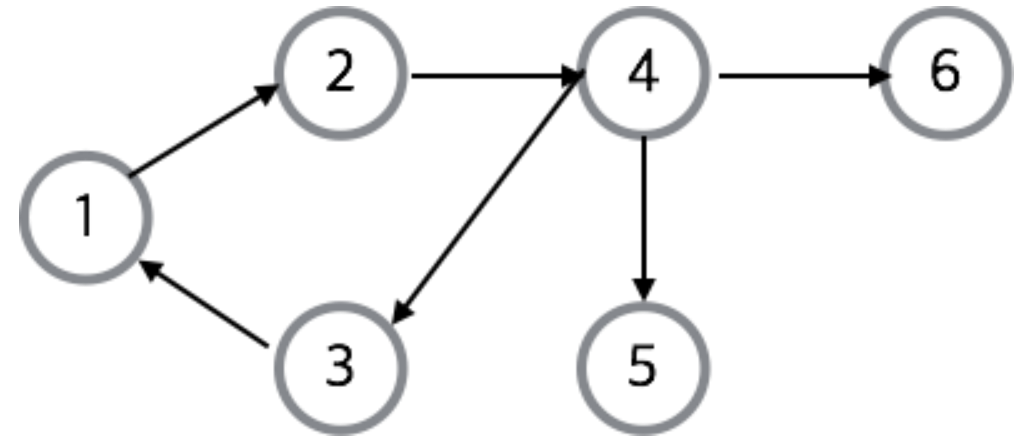
트리



스택



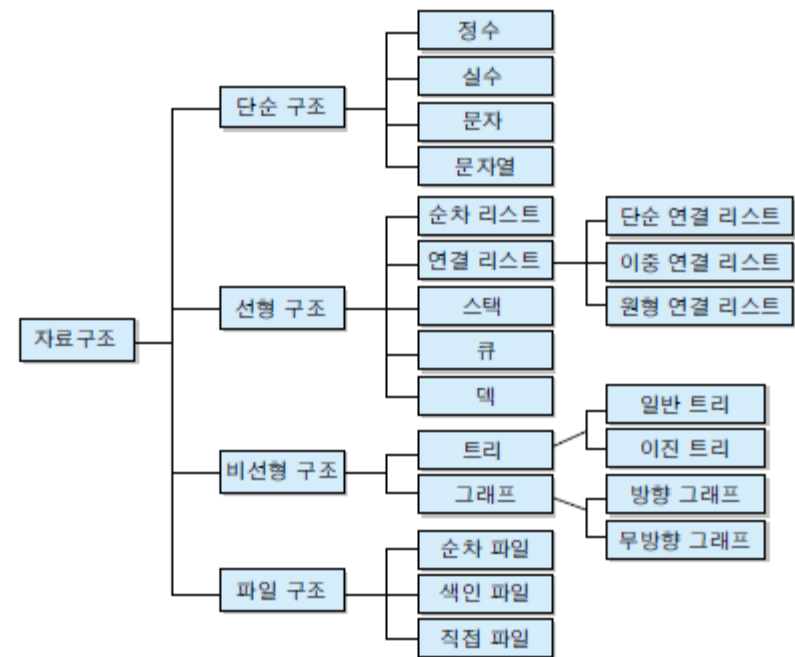
큐



그래프

자료구조 분류

자료구조의 분류는 다음과 같습니다.
본 스터디에서는 스택, 큐, 덱, 트리,
그래프를 다룹니다!



스택 큐 덱

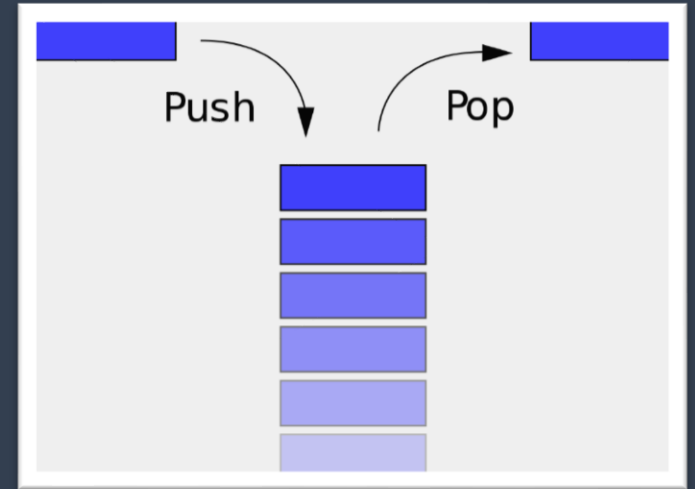
이번 시간에는 위 세가지 선형 자료구조를 배워봅시다!

스택

Stack

스택 Stack

한 쪽 끝에서만 자료를 넣거나 뺄 수 있는 구조입니다.
LIFO (Last In First Out)

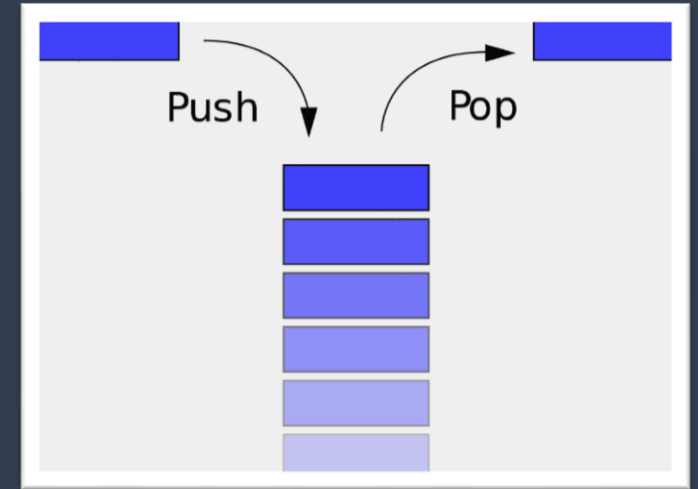


스택 Stack

한 쪽 끝에서만 자료를 넣거나 뺄 수 있는 구조입니다.

LIFO (Last In First Out)

사용 사례) 재귀 알고리즘, 웹 브라우저 방문기록
실행 취소, 역순 문자열 만들기, 수식의 괄호 검사
후위 표기법 계산, 호출 스택



스택 Stack

스택을 정의하려면 어떤 메소드들이 필요할까요?
한번 생각해봅시다!

스택 Stack

스택을 정의하기 위해서
다음 메소드들이 필요합니다!



1. **POP** 스택에서 자료를 빼기
2. **PUSH** 스택에 자료를 넣기
3. **EMPTY** 스택이 비어 있는지 확인
4. **TOP** 스택 최상단의 자료를 보는 연산
5. **SIZE** 스택에 저장된 데이터의 개수를 반환

스택 Stack

이들을 직접 구현하는 방법은
자료구조 수업시간에 배웁니다.

이번 스터디에서는 STL을 사용합니다!

STL

Standard Template Library 표준 C++ 라이브러리

스택, 큐, 벡터 등 프로그램에 필요한 자료구조와 알고리즘을
템플릿으로 제공하는 라이브러리입니다.

#본 스터디에서 자주 사용하게 될 친구입니다!

스택 Stack

한번 STL을 사용하여 스택을 구현해봅시다!

<https://twpower.github.io/75-how-to-use-stack-in-cpp>

#위 링크를 참고해주세요! 매우 간단합니다!



연습 문제
10828번 스택

<https://www.acmicpc.net/problem/10828>

STL을 사용하여 간편하게 구현 할 수 있습니다!



연습 문제

9093번 단어 뒤집기

<https://www.acmicpc.net/problem/9093>

스택의 특징을 활용하면 간단하게 풀 수 있습니다!



연습 문제
9012번 괄호

<https://www.acmicpc.net/problem/9012>

스택을 이용한 문제입니다.
우선 먼저 한번 도전해보세요!
그 후 다음 슬라이드에서 같이 접근해봅시다!

우선 문제를 이해해봅시다!

문제에 이런저런 설명이 많지만
결국은 올바른 괄호 문자열인지 판단하는 문제입니다.
이처럼 알고리즘 문제 설명에서 필요한 정보만을 뽑아내는 것이 중요합니다!

문제를 어떻게 접근해야 할까요?

컴퓨터가 풀 수 있다는 뜻은
사람도 풀 수 있다는 뜻입니다.
예제 입력들을 직접 손으로 풀어보며
문제 해결 아이디어를 얻어봅시다!

만약 예제 입력이 까다롭다면
직접 간단한 예제 입력을 만들어봅시다.

문제를 어떻게 접근해야 할까요?

(()) ()

다음 문자열은 왜 올바른 괄호 문자열이 아닐까요?
어떤 문자열이 올바른 괄호 문자열인가요?
그렇다면 올바른 괄호 문자열은 어떻게 정의할 수 있을까요?

문제를 어떻게 접근해야 할까요?

(()) (())

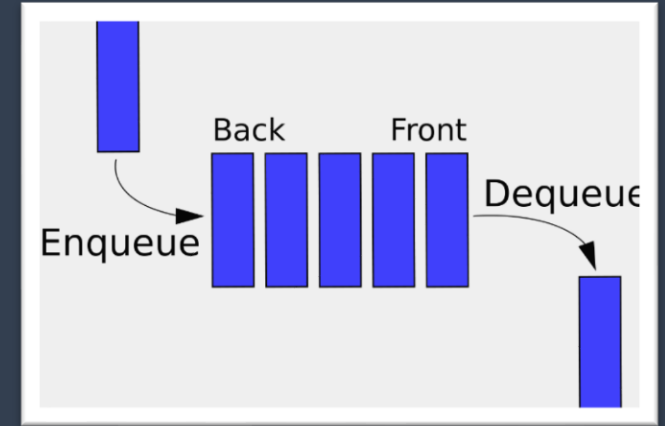
TBA

큐

Queue

큐 Queue

한 쪽 끝에서만 자료를 넣고
다른 한 쪽 끝에서만 뺄 수 있는 구조입니다.
LIFO (Last In Last Out)

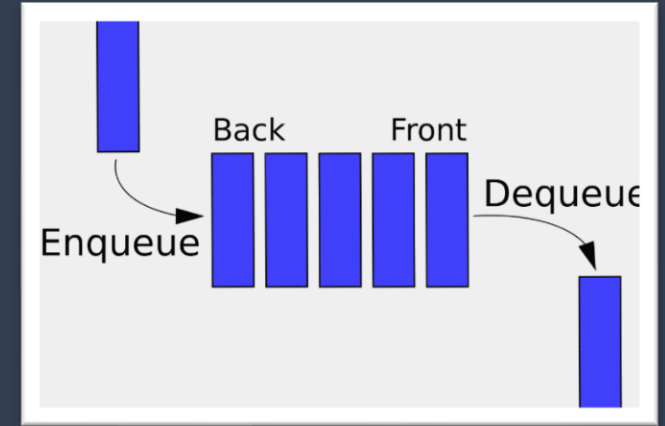


큐 Queue

한 쪽 끝에서만 자료를 넣고
다른 한 쪽 끝에서만 뺄 수 있는 구조입니다.

FIFO(First In First Out)

사용 사례) 연산 처리 작업 대기, 프린트 출력 문서 대기
컨텐츠 버퍼링, 대기열



큐 Queue

큐를 정의하기 위해서
다음 메소드들이 필요합니다!



1. **POP** 큐에서 자료를 빼기
2. **PUSH** 큐에 자료를 넣기
3. **EMPTY** 스택이 비어 있는지 확인
4. **FRONT** 큐 가장 앞의 자료를 보는 연산
5. **BACK** 큐 가장 뒤의 자료를 보는 연산
6. **SIZE** 큐에 저장된 데이터의 개수를 반환

큐 Queue

이번에도 STL을 사용하여 큐를 구현해봅시다!

<https://twpower.github.io/76-how-to-use-queue-in-cpp>

#위 링크를 참고해주세요! 매우 간단합니다!



연습 문제
10845번 큐

<https://www.acmicpc.net/problem/10845>

스택과 마찬가지로 STL을 사용하면
간단하게 구현할 수 있습니다.



연습 문제

1158번 요세푸스 문제

<https://www.acmicpc.net/problem/1158>

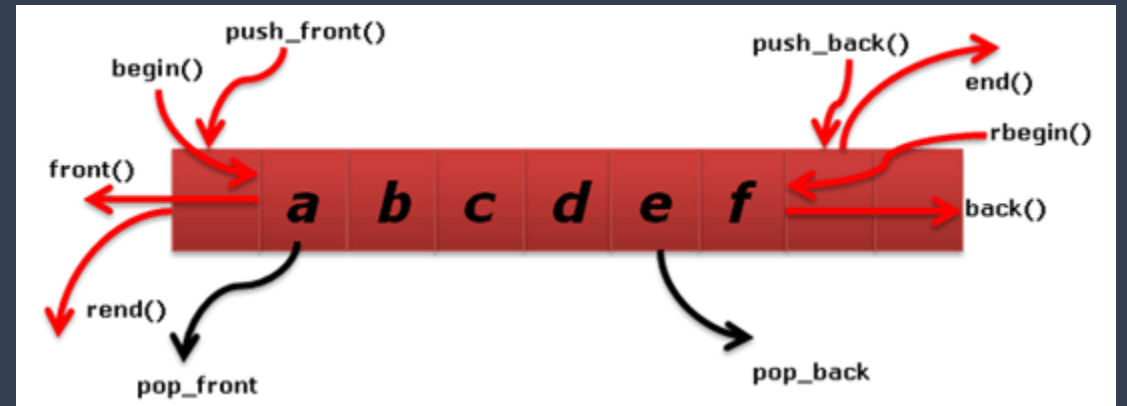
문제만 잘 이해하면 쉽게 해결 할 수 있습니다!

덱

Deque

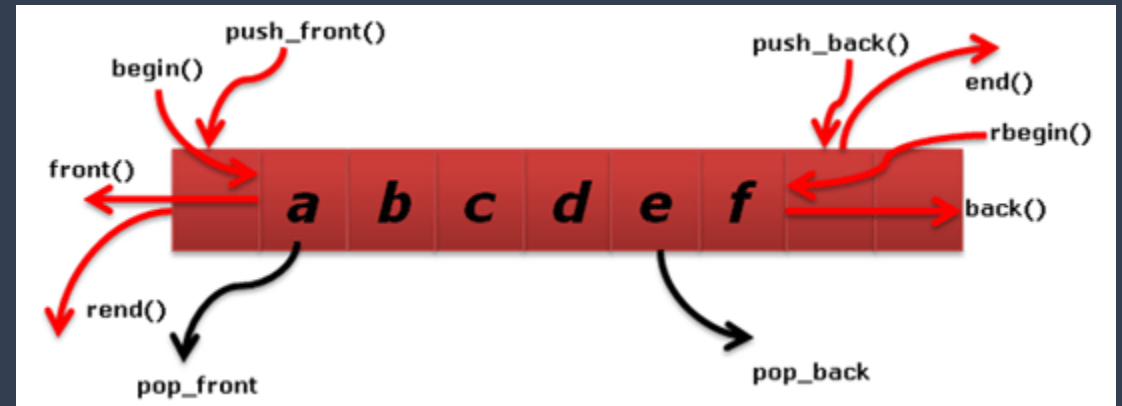
덱 Deque

양쪽에서 자료를 넣고 뺄 수 있는 구조입니다.
스택과 큐의 확장형이라 생각하면 편합니다.



덱 Deque

양쪽에서 자료를 넣고 뺄 수 있는 구조입니다.
스택과 큐의 확장형이라 생각하면 편합니다.
사용사례) 스케줄링



큐 Queue

덱 또한 STL이 존재합니다!

<https://blockdmask.tistory.com/73>

#위 링크를 참고해주세요! 매우 간단합니다!



연습 문제
10866번 덱

<https://www.acmicpc.net/problem/10866>

앞서 배운 STL을 활용하여 문제를 해결해 봅시다!

정리

이번 시간에는

스택 큐 덱에 대해서 배웠습니다.

세 자료구조들은 간단하지만 앞으로 배울 그래프, 트리 등의

자료구조를 구현할 때 기초가 됩니다.

또한 컴퓨터 과학에서도 CPU, 메모리, 프로그램 동작과 연산에서

가장 기초적이고 중요하게 사용되는 자료구조입니다.

반드시, 꼭! 스택과 큐 그리고 덱의 개념은 기억해두시기 바랍니다!

The End

수고하셨습니다!

수학의 꽃이 미적분이라면

PS의 꽃은 DP입니다.

다음시간에는

Dynamic Programming을 배워봅시다.