Lab. Session 1 - Assign 0,1

Computer Security Lab.

Name: Sangyun Kim

Email: sangyun.kim@snu.ac.kr

Contents

Contents

00-Setup Goal

- 자료구조의 기초 과목 과제의 구성을 이해한다.

- C++ 코드 컴파일과 관련된 tool을 이해한다. (g++, make, cmake)

- 기본적인 C++ 문법 (+ 조금의 Modern C++ 문법)에 대해 복습/학습한다.

Repository Clone

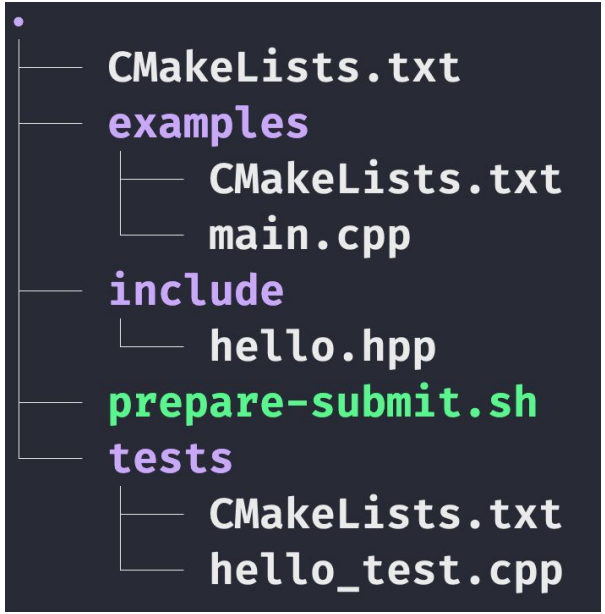
- 과제 Repository
  - https://github.com/compsec-snu-class/data-structures-23fall
  - 과제 관련 코드는 위 링크에 업데이트 됨

- Containter 접속 후, 아래 명령어를 통해 Repository Clone
  - git clone https://github.com/compsec-snu-class/data-structures-23fall.git

```
Cloning into 'data-structures-23fall'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 36 (delta 5), reused 32 (delta 1), pack-reused 0
Receiving objects: 100% (36/36), 10.58 KiB | 5.29 MiB/s, done.
Resolving deltas: 100% (5/5), done.
compsec@9612f5988f71:~$ ls
data-structures-23fall
```

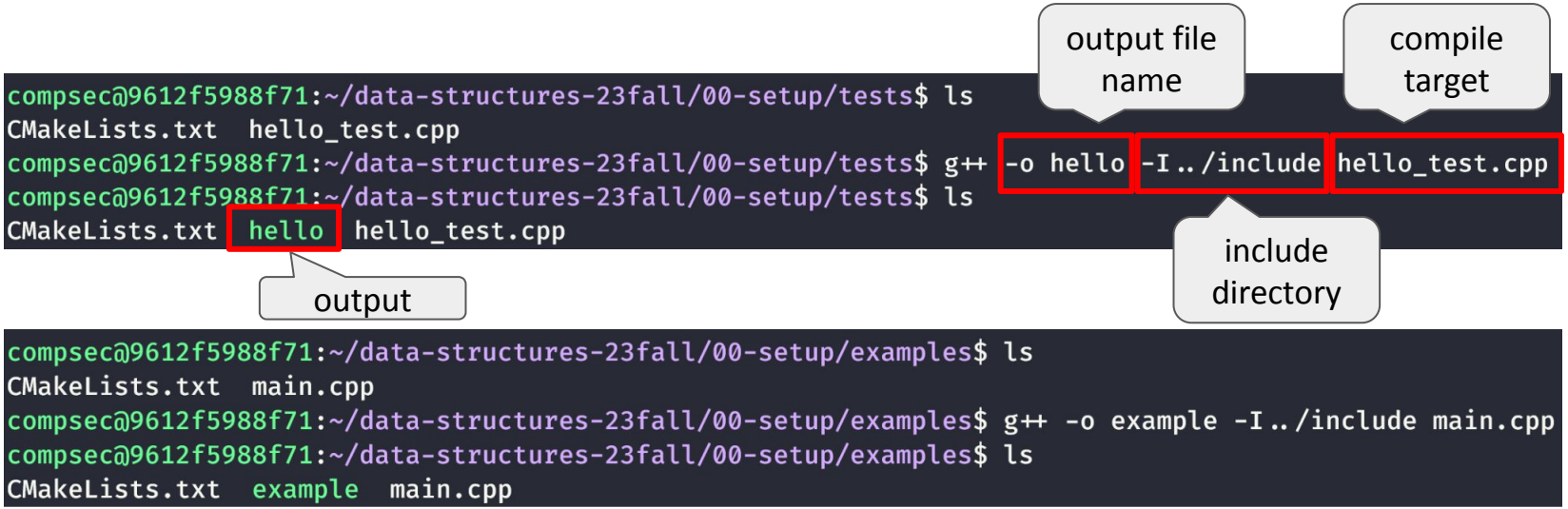
과제 Directory 구조

- examples
  - 예제 코드
  - 작성한 자료구조를 테스트 해 볼 수 있음
- include
  - 자료구조 관련 파일
  - 대부분의 과제에서 작성해야 할 코드는 해당 디렉토리에 위치
- tests
  - 채점 관련 코드
- prepare-submit.sh
  - 작성한 코드를 tar 압축파일로 압축



C++ 코드 컴파일

- Compiler: g++ (GNU C++ Compiler)



C++ 기초 문법

- 수업시간에 배운 기초 C++은 숙지하고 있어야 과제하는데 어려움이 없습니다.
  - control statement
  - operators
  - arrays
  - functions
  - namespace
  - classes (constructor, destructor, virtual function, etc…)
  - templates
  - pointers (*, &)
  - memory allocation
  - operator overloading
  - lvalue reference (&)
  - 등등…

Modern C++?

- C++도 계속해서 발전합니다.
  - 3년에 한 번씩, 새로운 C++ 표준안이 발표됩니다. (C++11, C++14, C++17, … , C++26)
  - C++11 이후의 C++ 표준을 Modern C++으로 칭합니다.

- 새로운 문법 및 라이브러리가 많이 등장합니다.
  - **auto**
  - Type alias, alias template
  - Range-based for
  - lambda expression
  - **<memory> library (unique_ptr, shared_ptr, etc…)**
  - **optional**
  - rvalue reference (&&)

Contents

- 00 - setup

- 01 - deque

01-Deque Goal

- Get familiar with C++ programming by implementing simple data structures like (expandable) arrays and linked lists.

- Get to know how unit testing works and how the rest of the assignments will be graded.

- Write a simple algorithm on top of the data structures that you wrote.
  Learn how to separate the interfaces (deque) from the implementations (array or linked list).

Deque

- Accroding to cppreference.com
  std::deque (double-ended queue) is an **indexed sequence container** that allows **fast insertion and deletion at both its beginning and its end**.

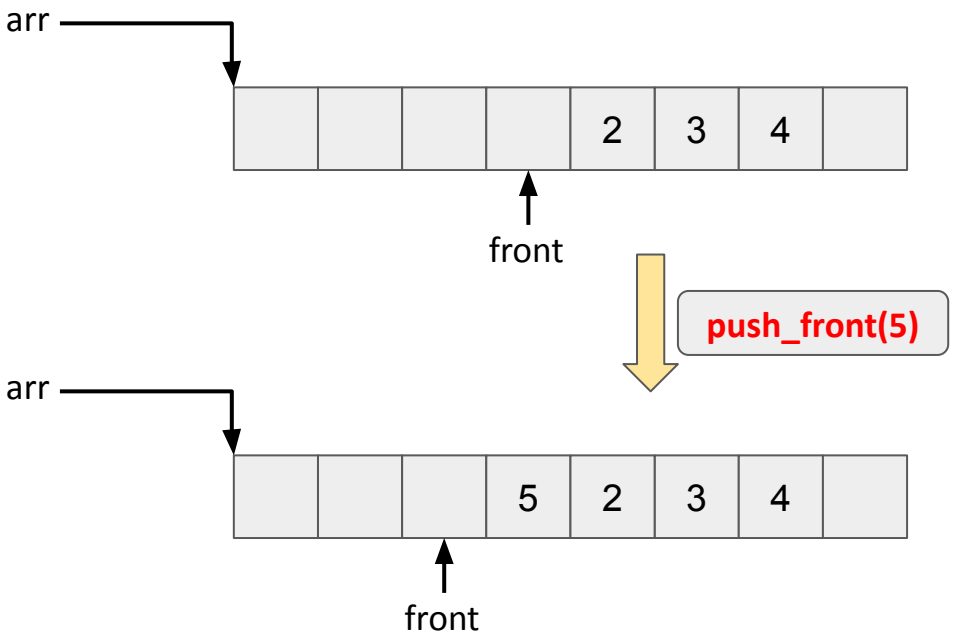
Deque Interface

```cpp
template <typename T>
class Deque {
public:
    virtual ~Deque() = default;

    /* NOTE: We won't implement push functions that take rvalue references. */
    virtual void push_front(const T&) = 0;
    virtual void push_back(const T&) = 0;

    /* NOTE: Unlike STL implementations which have separate `front` and
       pop_front` functions, we have one unified method for removing an elem. */
    virtual std::optional<T> remove_front() = 0;
    virtual std::optional<T> remove_back() = 0;

    virtual bool empty() = 0;
    virtual size_t size() = 0;

    virtual T& operator[](size_t) = 0;
};
```

Deque Implementation

- 이번 과제에서는, 2가지 수단으로 Deque을 구현합니다.
  - Array
  - Double Linked List

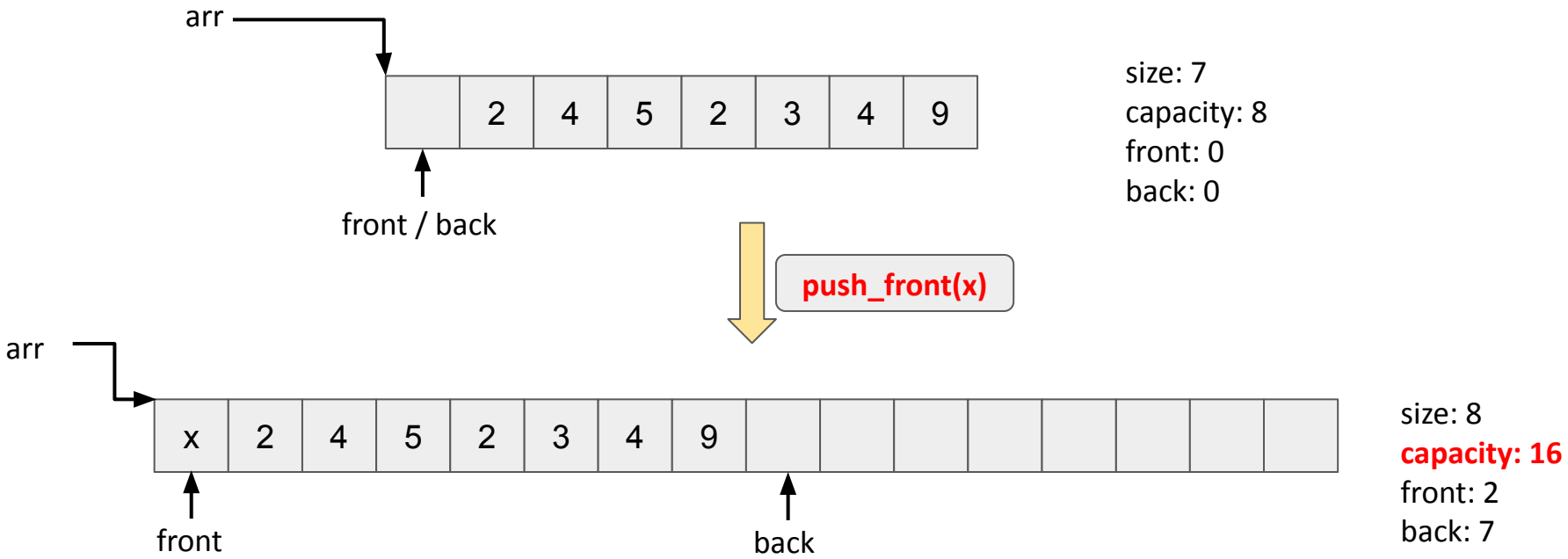
Deque Implementation - Array



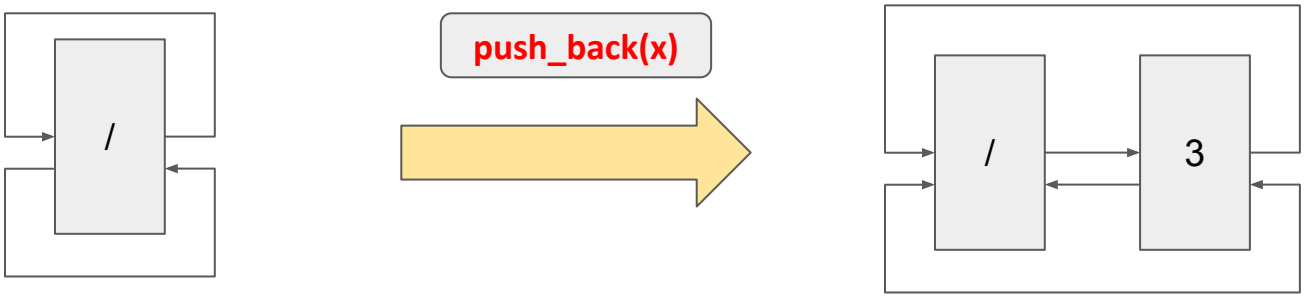
size: 3
capacity: 8
front: 3
back: 7

size: 4
capacity: 8
front: 2
back: 7

Deque Implementation - Array



Deque Implementation - Double Linked List



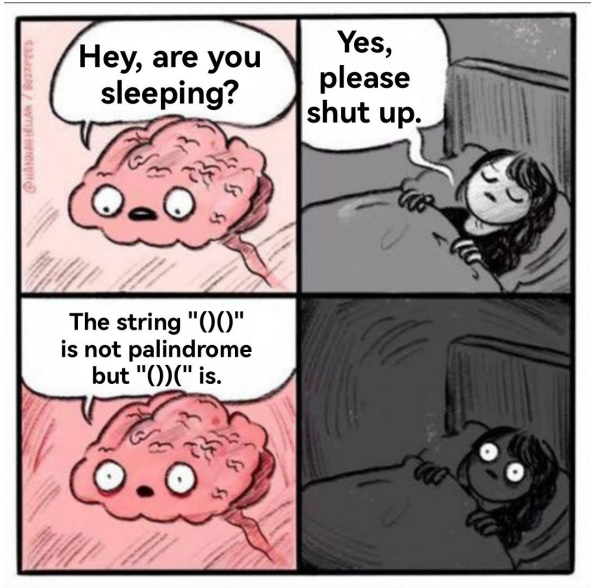
- Array 구현과 달리 doubling 과정이 필요 없음
- next / prev pointer를 통하여 관리
- 이번 과제에서는 raw pointer를 사용합니다.
  - 이후 과제에서는 smart pointer를 사용합니다.

Deque Application - Palindrome

- 구현한 Deque를 이용하여 Application을 해결해봅시다.

- What's the palindrome?
  - 회문: 앞으로 읽으나 뒤로 읽으나 똑같은 단어

```
template<typename Deque>
bool Palindrome<Deque>::is_palindrome(const std::string& s1) {
    // TODO
    return false;
}
```



Memory Leak

- 모든 과제는 Memory Leak 테스트를 통과해야 합니다.

- Memory Leak 이란?
  - 사용하지 않을 메모리를 계속 점유하는 상황
  - Memory Leak이 지속적으로 유지되는 경우 시스템 메모리가 부족해질 수 있음

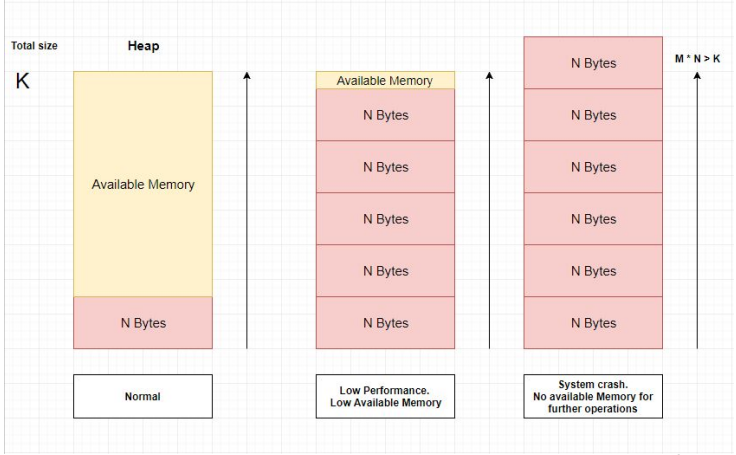


image ref:https://www.log2base2.com/C/pointer/free-in-c.html

Grading

- 아래 두 항목을 모두 통과하면 만점입니다.

- test 케이스 전부 통과
  - 과제에 따라 몇몇 테스트는 공개되지 않는 경우도 있습니다.
  - 항상 Edge case를 생각하며 코딩하시고, 직접 테스트를 해보시기 바랍니다.

- Memory Leak 테스트 통과

Assignment Due

- 10월 3일 화요일 11:59 PM

- 제출 방법: 제출 서버에 업로드
  - 제출 서버 주소는 추후 공지

- Delay 제출
  - 추후 공지

마지막으로...

- 아무리 사소한 것이라도 모르면 질문해주세요
  - 단, 반드시 인터넷에 검색을 먼저 해본 후에 그래도 모르겠다면 질문해주세요.

- eTL에 질문글을 올릴 때, 가급적 공개글로 해주세요
  - 모르는 것은 부끄러운 것이 아닙니다.
  - 공개된 질의응답이 다른 학생들에게도 큰 도움이 됩니다.