STL 기초

Stack / Queue

한컴에듀케이션

STL Container

- **Sequence**
  - array : static array
  - vector : dynamic array
  - deque : dynamic array
  - foward_list : singly linked list
  - list : doubly linked list

- **Adaptors**
  - **stack** : LIFO
  - **queue** : FIFO
  - priority_queue : 우선순위 큐

- **Associative** (Red-Black Tree)
  - set : (Key) 중복X
  - multiset : (Key) 중복O
  - map : (Key,Value), 중복X
  - multimap : (Key,Value), 중복O

- **Unordered associative** (Hash)
  - unordered_set : (Key) 중복X
  - unordered_multiset : (Key) 중복O
  - unordered_map : (Key,Value), 중복X
  - unordered_multimap : (Key,Value), 중복O

Adaptors

sequence container 활용하여 stack, queue, priority_queue interface 제공

- Iterator 제공 X
- 중간 원소 접근 X

  stack : top only

  queue : front, back only

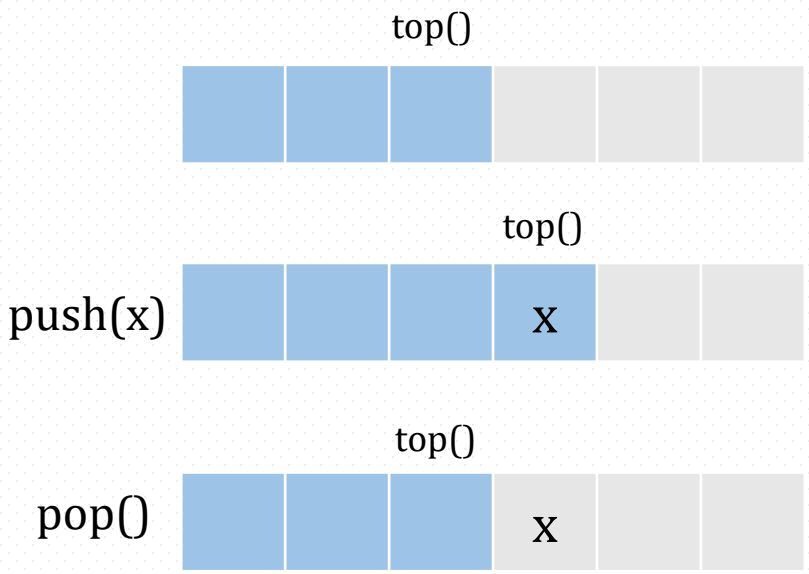
  priority_queue : top only

- **stack** : 거의 쓸 일 없음
- **queue** : BFS에서 주로 사용
- **priority_queue**
  - 빈번한 삽입, 삭제 속에서 우선순위 구하는 경우에 사용
  - set, map에 비해 활용성 떨어짐
  - 정렬기준 가장 오른쪽 원소가 top으로 올라옴
  - default : max pq

stack

- #include<stack>
- Last Input, First out
- 항상 top만 접근 가능

```
template<
    class T,
    class Container = std::deque<T>
> class stack;
```



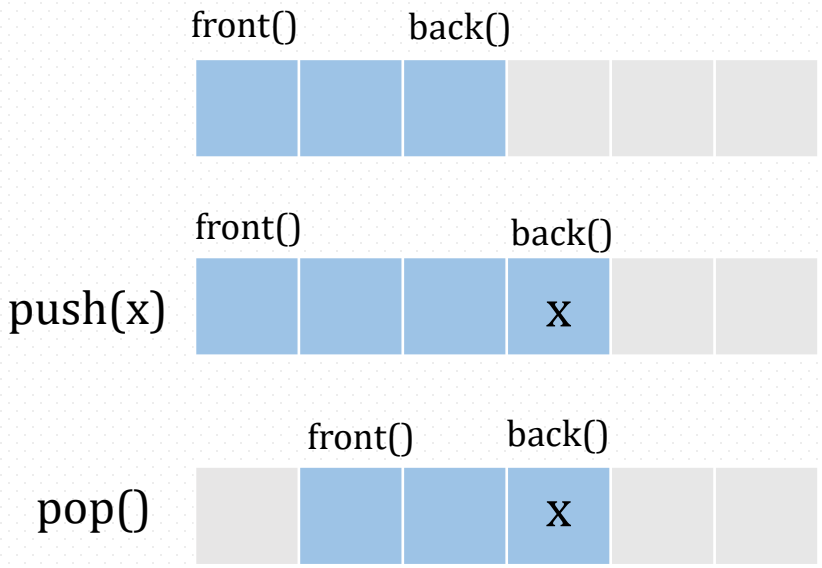

```
stack<T> st
st = {} // clear

void st.push(T)
void st.pop()
T    st.top()
int  st.size()
bool st.empty()
```

queue

- #include<queue>
- First Input, First out
- 항상 front, back만 접근 가능

```
template<
    class T,
    class Container = std::deque<T>
> class queue;
```




```
queue<T> q
q = {}// clear

void q.push(x)
void q.pop()
T    q.front()
T    q.back()
int  q.size()
bool q.empty()
```


감사합니다