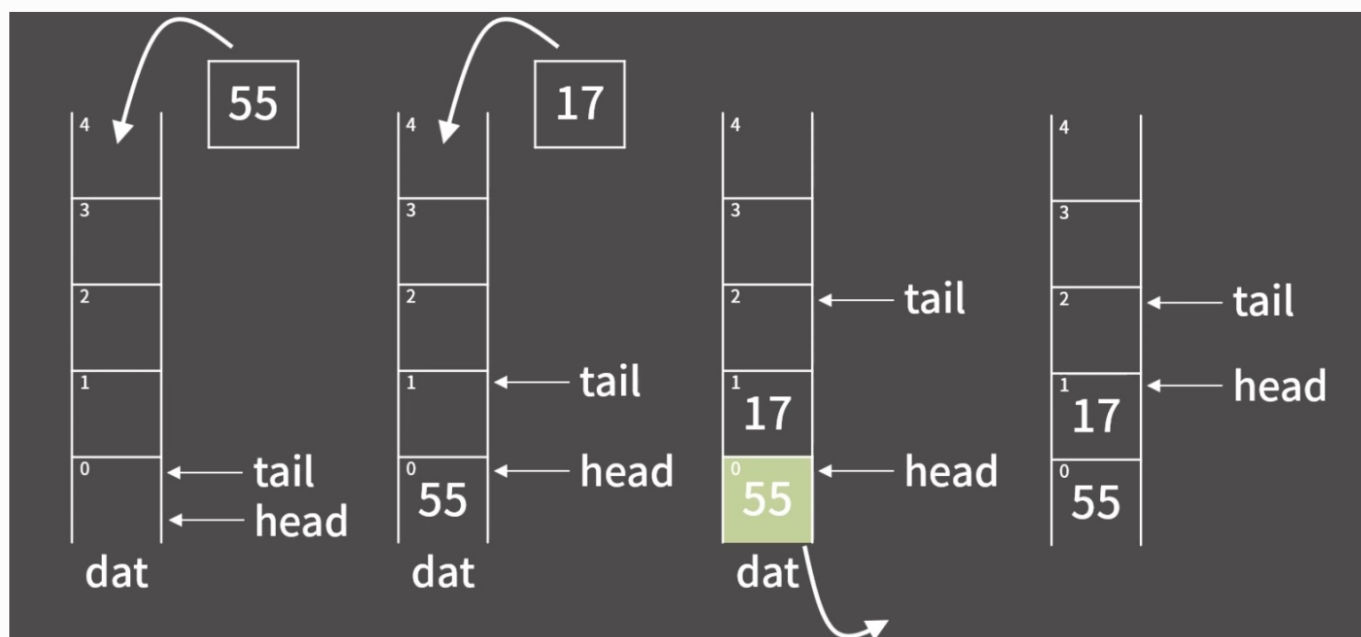


1. 원소의 추가가 $O(1)$
2. 원소의 제거가 $O(1)$
3. 제일 앞/뒤의 원소 확인이 $O(1)$
4. 제일 앞/뒤가 아닌 나머지 원소들의 확인/변경이 원칙적으로 불가능

구현



pop할 때 배열을 한칸씩 앞으로 당기지 않고 그냥 head를 +1한다
 이런 메모리가 낭비되는데 원형 큐를 이용해서 해결할 수 있다
 큐의 크기: $tail - head$



head나 tail이 7이면 0이 되도록 하면 된다
 근데 컨테이너에서는 어차피 최대 push가 256까지 있어서
 선형 큐로 해도 된다

STL queue

```
#include <bits/stdc++.h>
using namespace std;

int main(void) {
    queue<int> Q;
    Q.push(10); // 10
    Q.push(20); // 10 20
    Q.push(30); // 10 20 30
    cout << Q.size() << '\n'; // 3
    if(Q.empty()) cout << "Q is empty\n";
    else cout << "Q is not empty\n"; // Q is not empty
    Q.pop(); // 20 30
    cout << Q.front() << '\n'; // 20
    cout << Q.back() << '\n'; // 30
    Q.push(40); // 20 30 40
    Q.pop(); // 30 40
    cout << Q.front() << '\n'; // 30
}
```

Queue가 비어있을 때 front, back, pop 호출시
 런타임에러 조심

연습문제 10845

그냥 구현이라 패스