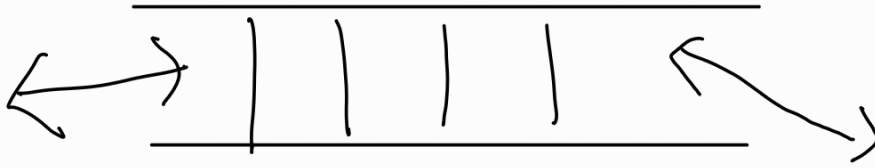


Deque : double-ended queue

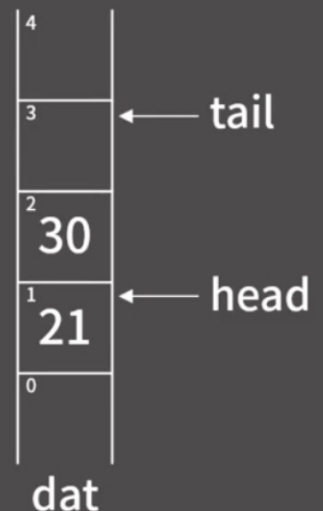


1. 원소의 추가가 $O(1)$
2. 원소의 제거가 $O(1)$
3. 제일 앞/뒤의 원소 확인이 $O(1)$
4. 제일 앞/뒤가 아닌 나머지 원소들의 확인/변경이 원칙적으로 불가능

↳ STL deque에서는 독특하게 인덱스로 원소에 접근 가능
(STL stack, queue ... 불가능)

```
01 const int MX = 1000005;  
02 int dat[2*MX+1];  
03 int head = MX, tail = MX;
```

덱은 양쪽에서 push, pop이 일어나서
중간을 처음 지점으로



size : tail - head

STL deque

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

int main(void){
    deque<int> DQ;
    DQ.push_front(10); // 10
    DQ.push_back(50); // 10 50
    DQ.push_front(24); // 24 10 50
    for(auto x : DQ) cout<<x;
    cout << DQ.size() << '\n'; // 3
    if(DQ.empty()) cout << "DQ is empty\n";
    else cout << "DQ is not empty\n"; // DQ is not empty
    DQ.pop_front(); // 10 50
    DQ.pop_back(); // 10
    cout << DQ.back() << '\n'; // 10
    DQ.push_back(72); // 10 72
    cout << DQ.front() << '\n'; // 10
    DQ.push_back(12); // 10 72 12
    DQ[2] = 17; // 10 72 17
    DQ.insert(DQ.begin()+1, 33); // 10 33 72 17
    DQ.insert(DQ.begin()+4, 60); // 10 33 72 17 60
    for(auto x : DQ) cout << x << ' ';
    cout << '\n';
    DQ.erase(DQ.begin()+3); // 10 33 72 60
    cout << DQ[3] << '\n'; // 60
    DQ.clear(); // DQ의 모든 원소 제거
}
```

double-ended queue라는 느낌보다 vector와 비슷함

vector와 달리 front push, pop도 $O(1)$

vector와 달리 deque은 원소들이 메모리 상에 연속하게 배치되어 있지 않다

=> front push, pop are required ? deque : vector

연습문제 10866

그냥 구현이라 패스