



# 실전 알고리즘 0x06강

## 큐

BaaaaaaaaaaaaaaaaarkingDog

# 목차



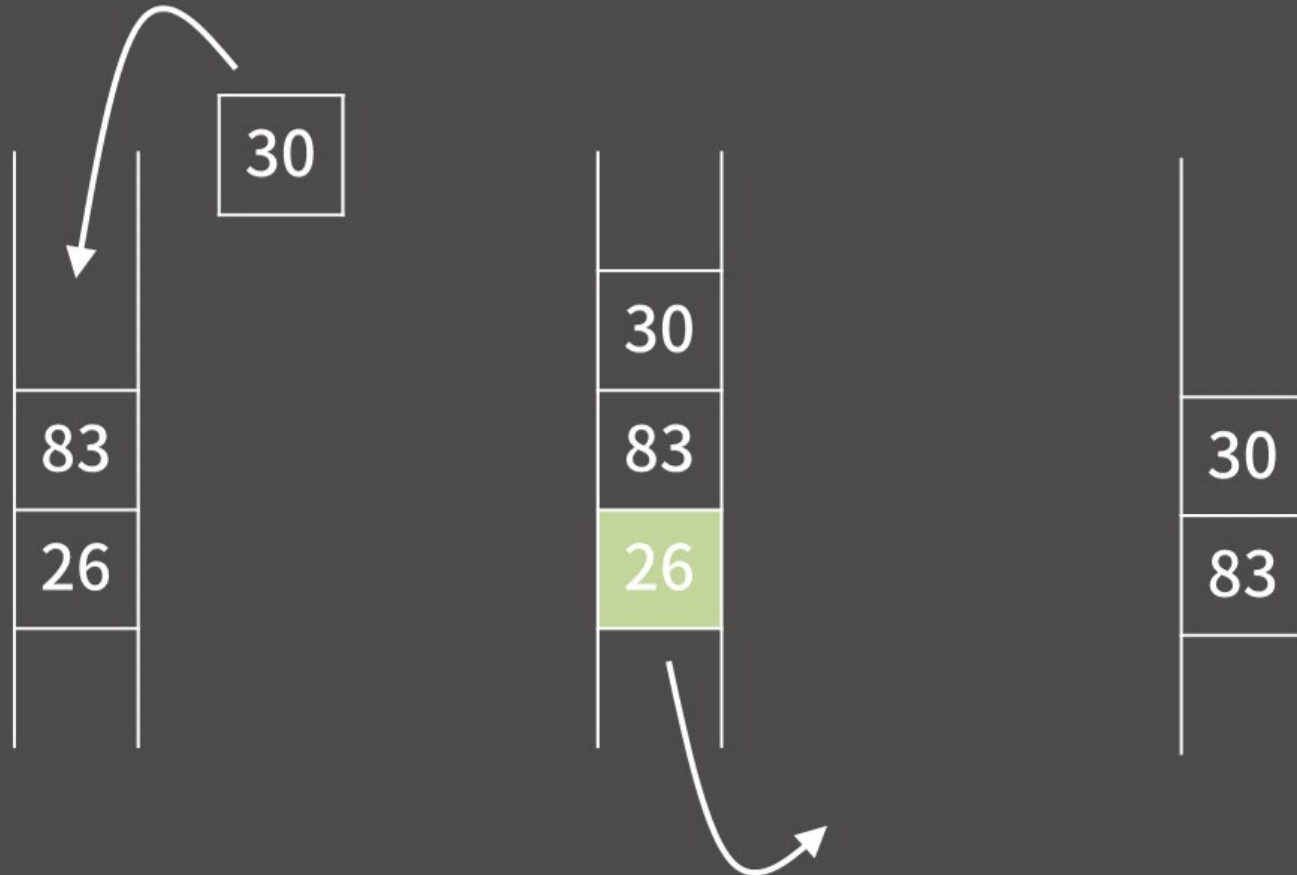
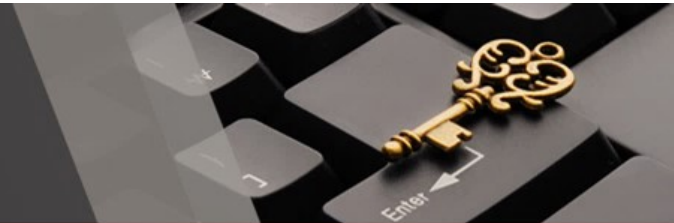
0x00 정의와 성질

0x01 기능과 구현

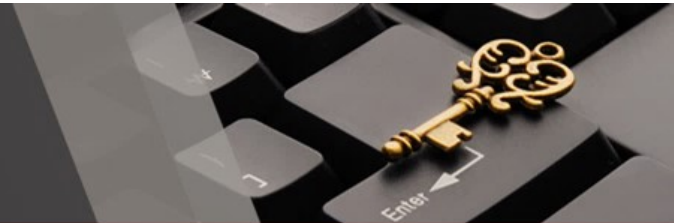
0x02 STL queue

0x03 연습문제

# 0x00 정의와 성질



# 0x00 정의와 성질



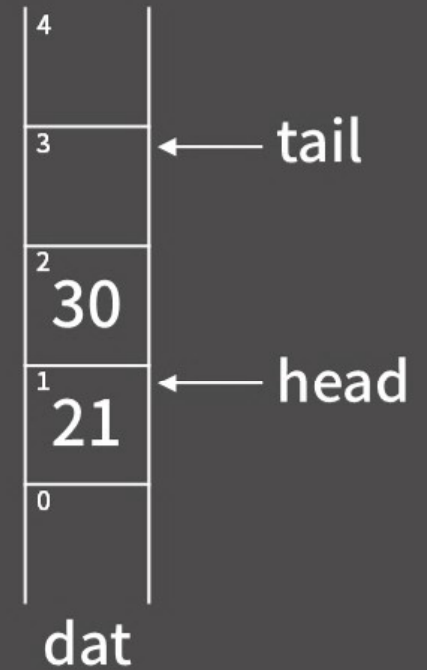
## 큐의 성질

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

# 0x01 기능과 구현

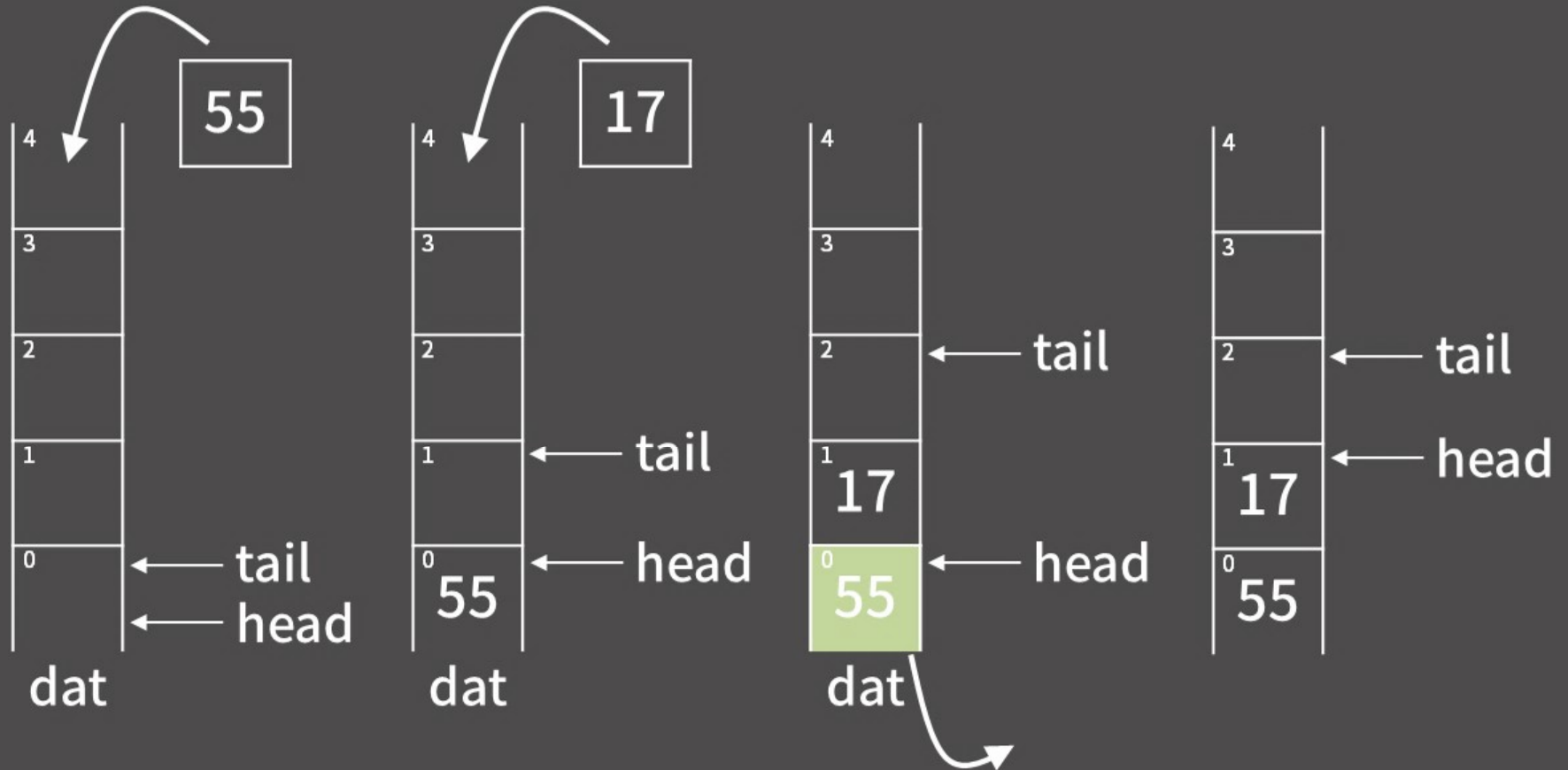
## 구현

```
01  const int MX = 1000005;  
02  int dat[MX];  
03  int head = 0, tail = 0;
```



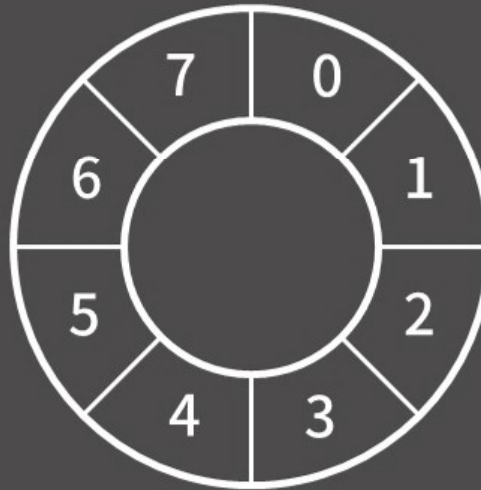
# 0x01 기능과 구현

구현



# 0x01 기능과 구현

구현



# 0x01 기능과 구현

## 구현

[https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/queue\\_test.cpp](https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/queue_test.cpp)

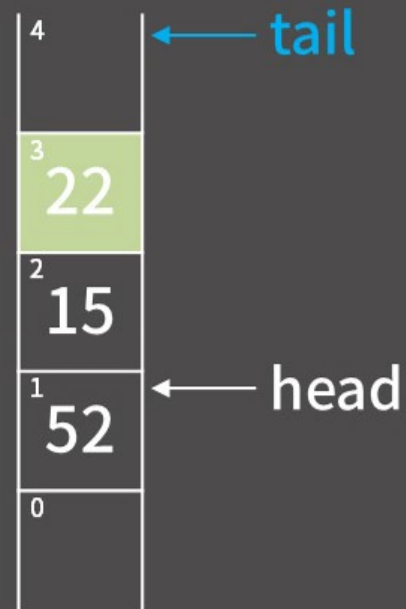
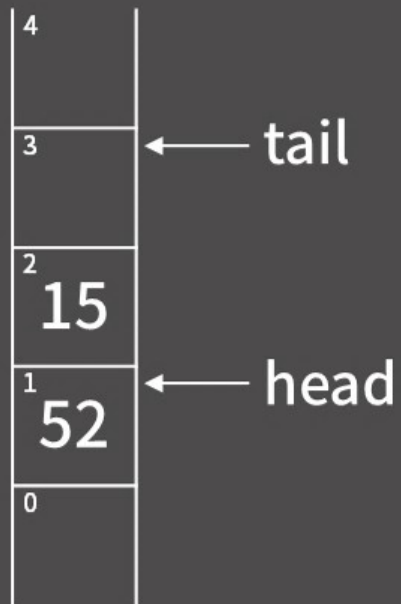
```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  const int MX = 1000005;
05  int dat[MX];
06  int head = 0, tail = 0;
07
08  void push(int x) {
09
10  }
11
12  void pop() {
13
14  }
```

```
15  int front() {
16
17  }
18
19  int back() {
20
21  }
22
23  void test() {
24      ...
25  }
26
27  int main(void) {
28      test();
29  }
```



# 0x01 기능과 구현

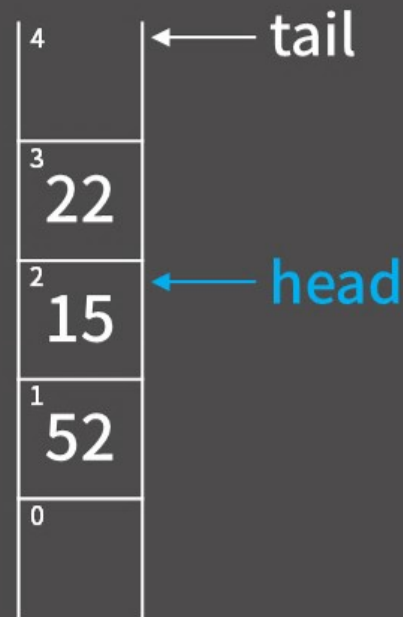
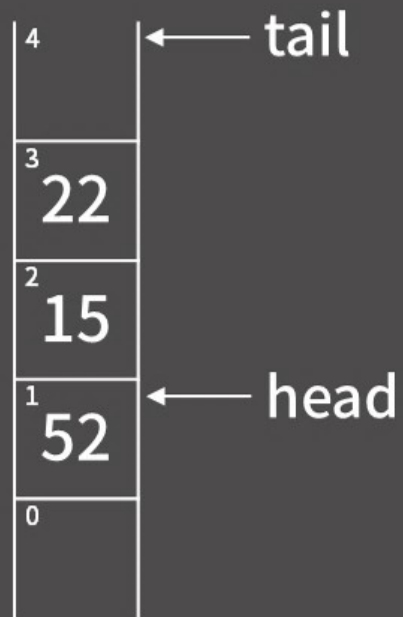
## push 함수



```
01 void push(int x) {  
02     dat[tail++] = x;  
03 }
```

# 0x01 기능과 구현

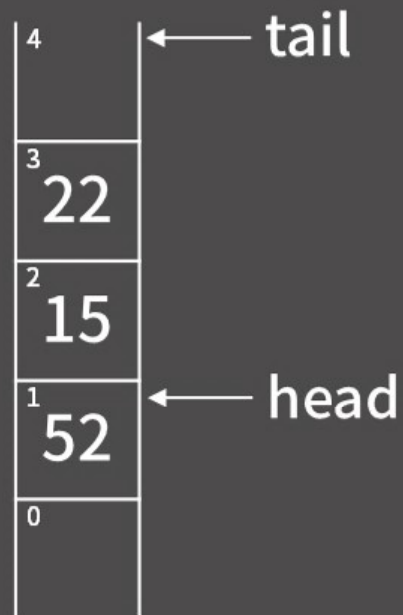
## pop 함수



```
01 void pop() {  
02     head++;  
03 }
```

# 0x01 기능과 구현

## front/back 함수



```
01 int front() {  
02     return dat[head];  
03 }  
04  
05 int back() {  
06     return dat[tail-1];  
07 }
```

[https://github.com/blisstoner/basic-algo-lecture-material/  
blob/master/0x06/queue\\_test\\_ans.cpp](https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/queue_test_ans.cpp)

# 0x02 STL queue

reference : <http://www.cplusplus.com/reference/queue/queue/>

[https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/queue\\_example.cpp](https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/queue_example.cpp)

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int main(void) {
05      queue<int> Q;
06      Q.push(10); // 10
07      Q.push(20); // 10 20
08      Q.push(30); // 10 20 30
09      cout << Q.size() << '\n'; // 3
10      if(Q.empty()) cout << "Q is empty\n";
11      else cout << "Q is not empty\n"; // Q is not empty
12      Q.pop(); // 20 30
13      cout << Q.front() << '\n'; // 20
14      cout << Q.back() << '\n'; // 30
15      Q.push(40); // 20 30 40
16      Q.pop(); // 30 40
17      cout << Q.front() << '\n'; // 30
18  }
```

# 0x03 연습문제

## BOJ 10845번: 큐

[https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/10845\\_1.cpp](https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/10845_1.cpp)

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int main(void) {
05      ios::sync_with_stdio(0);
06      cin.tie(0);
07
08      queue<int> Q;
09      int n;
10      cin >> n;
11      while(n--){
12          string q;
13          cin >> q;
14          if(q=="push"){
15              int val;
16              cin >> val;
17              Q.push(val);
18          }
```

```
19      else if(q=="pop"){
20          if(Q.empty()) cout << -1 << '\n';
21          else{
22              cout << Q.front() << '\n';
23              Q.pop();
24          }
25      }
26      else if(q=="size"){
27          cout << Q.size() << '\n';
28      }
29      else if(q=="empty"){
30          cout << Q.empty() << '\n';
31      }
32      else if(q=="front"){
33          if(Q.empty()) cout << -1 << '\n';
34          else cout << Q.front() << '\n';
35      }
36      else{ // back
37          if(Q.empty()) cout << -1 << '\n';
38          else cout << Q.back() << '\n';
39      }
40  }
41 }
```

# 0x03 연습문제

## BOJ 10845번: 큐

[https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/10845\\_2.cpp](https://github.com/blisstoner/basic-algo-lecture-material/blob/master/0x06/10845_2.cpp)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 const int MX = 1000005;
05 int dat[MX];
06 int head, tail;
07
08 void push(int x){
09     dat[tail++] = x;
10 }
11
12 void pop(){
13     head++;
14 }
15
16 int front(){
17     return dat[head];
18 }
```

```
19 int back(){
20     return dat[tail-1];
21 }
22
23 int main(void) {
24     ios::sync_with_stdio(0);
25     cin.tie(0);
26     int n;
27     cin >> n;
28     while(n--){
29         string q;
30         cin >> q;
31         if(q=="push"){
32             int val;
33             cin >> val;
34             push(val);
35         }
36         else if(q=="pop"){
37             if(tail==head)
38                 cout << -1 << '\n';
39             else{
40                 cout << front() << '\n';
41                 pop();
42             }
43         }
```

```
40         else if(q=="size"){
41             cout << tail-head << '\n';
42         }
43         else if(q=="empty"){
44             cout << (tail==head) << '\n';
45         }
46         else if(q=="front"){
47             if(tail==head)
48                 cout << -1 << '\n';
49             else cout << front() << '\n';
50         }
51         else{ // back
52             if(tail==head)
53                 cout << -1 << '\n';
54             else cout << back() << '\n';
55         }
56     }
57 }
58 }
```



# 강의 정리

