실전 알고리즘 0x07강

덱

BaaaaaaaaaaaaaaaarkingDog

목차

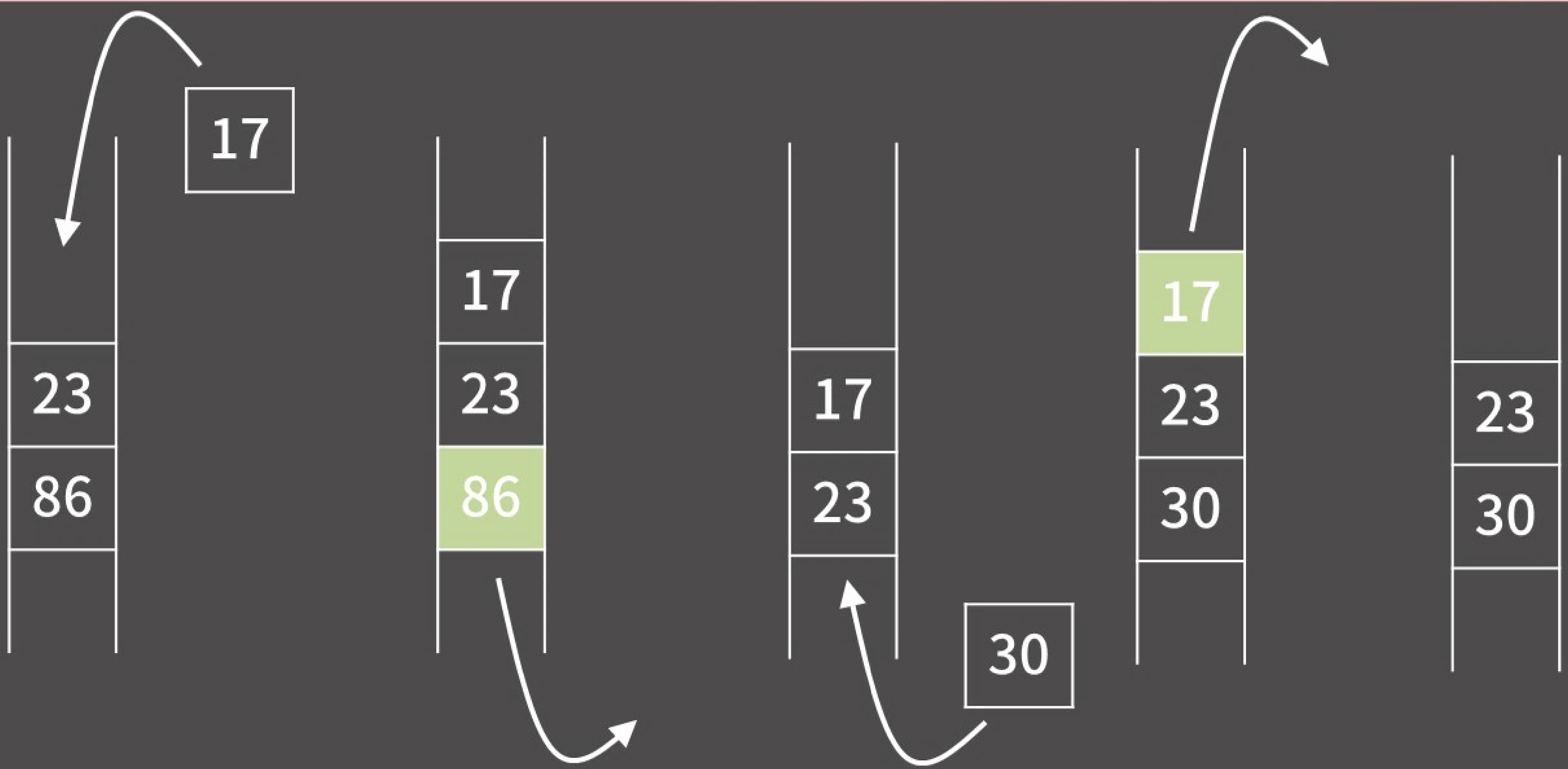
0x00 정의와 성질

0x01 기능과 구현

0x02 STL deque

0x03 연습문제

0x00 정의와 성질



17

23

86

17

23

86

17

23

30

17

23

30

23

30

0x00 정의와 성질

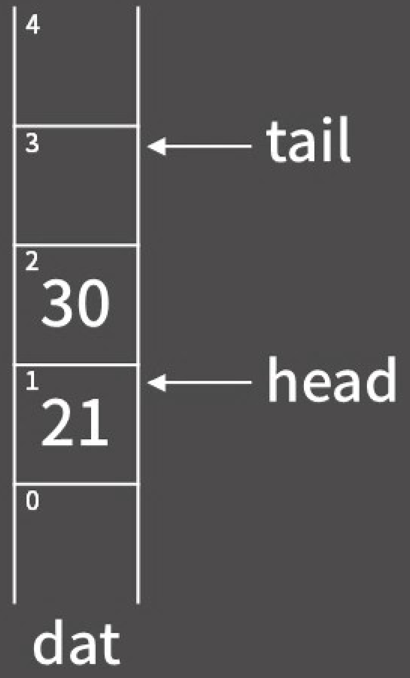
덱의 성질

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

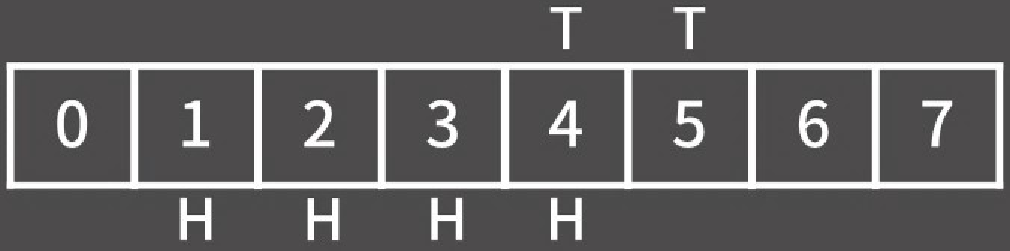
구현

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



0x01 기능과 구현

구현



				T	T		
0	1	2	3	4	5	6	7
	H	H	H	H			

0x01 기능과 구현

구현

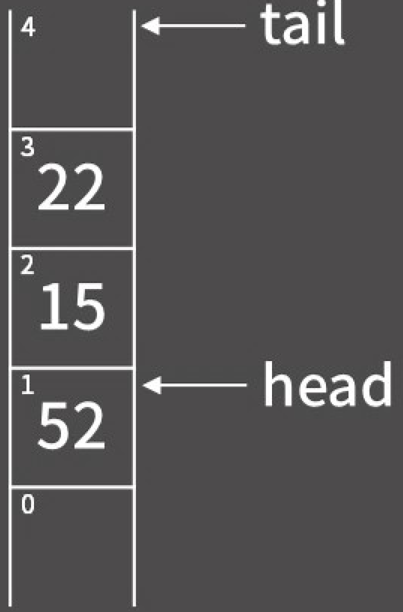
https://github.com/blisstoner/basic-algo-lecture-metarial/blob/master/0x07/deque_test.cpp

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

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

void push_front(int x){

}

void push_back(int x){

}

void pop_front(){

}
```

```cpp
void pop_back(){

}

int front(){

}

int back(){

}

void test(){
  ...
}

int main(void){
  test();
}
```

0x01 기능과 구현

구현



```cpp
void push_front(int x){
  dat[--head] = x;
}

void push_back(int x){
  dat[tail++] = x;
}

void pop_front(){
  head++;
}
```

```cpp
void pop_back(){
  tail--;
}

int front(){
  return dat[head];
}

int back(){
  return dat[tail-1];
}
```

https://github.com/blisstoner/basic-algo-lecture-metarial/blob/master/0x07/deque_test_ans.cpp

0x02 STL deque

reference : http://www.cplusplus.com/reference/deque/deque/

https://github.com/blisstoner/basic-algo-lecture-metarial/blob/master/0x07/deque_example.cpp

```cpp
#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
}
```

0x03 연습문제

BOJ 10866번: 덱

https://github.com/blisstoner/basic-algo-lecture-metarial/blob/master/0x07/10866_1.cpp

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

int main(void) {
  ios::sync_with_stdio(0);
  cin.tie(0);

  deque<int> DQ;
  int n;
  cin >> n;
  while (n--) {
    string q;
    cin >> q;
    if (q == "push_back") {
      int val;
      cin >> val;
      DQ.push_back(val);
    }
    else if (q == "push_front") {
      int val;
      cin >> val;
      DQ.push_front(val);
    }
    else if(q == "pop_front"){
      if(DQ.empty())
        cout << -1 << '\n';
      else{
        cout << DQ.front() << '\n';
        DQ.pop_front();
      }
    }
    else if(q == "pop_back"){
      if(DQ.empty())
        cout << -1 << '\n';
      else{
        cout << DQ.back() << '\n';
        DQ.pop_back();
      }
    }
    else if (q == "size")
      cout << DQ.size() << '\n';
    else if (q == "empty")
      cout << DQ.empty() << '\n';
    else if (q == "front") {
      if(DQ.empty())
        cout << -1 << '\n';
      else
        cout << DQ.front() << '\n';
    }
    else { // back
      if(DQ.empty())
        cout << -1 << '\n';
      else
        cout << DQ.back() << '\n';
    }
  }
}
```

0x03 연습문제

BOJ 10866번: 덱

https://github.com/blisstoner/basic-algo-lecture-metarial/blob/master/0x07/10866_2.cpp

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

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

...

int main(void) {
  ios::sync_with_stdio(0);
  cin.tie(0);
  int n;
  cin >> n;
  while (n--) {
    string q;
    cin >> q;
    if (q == "push_back") {
      int val;
      cin >> val;
      push_back(val);
    }
    else if (q == "push_front") {
      int val;
      cin >> val;
      push_front(val);
    }
    else if(q == "pop_front"){
      if(tail==head)
        cout << -1 << '\n';
      else{
        cout << front() << '\n';
        pop_front();
      }
    }
    else if(q == "pop_back"){
      if(tail==head)
        cout << -1 << '\n';
      else{
        cout << back() << '\n';
        pop_back();
      }
    }
    else if (q == "size")
      cout << DQ.size() << '\n';
    else if (q == "empty")
      cout << DQ.empty() << '\n';
    else if (q == "front") {
      if(DQ.empty())
        cout << -1 << '\n';
      else
        cout << DQ.front() << '\n';
    }
    else { // back
      if(DQ.empty())
        cout << -1 << '\n';
      else
        cout << DQ.back() << '\n';
    }
  }
}
```

강의 정리