

C언어 스터디

5.0주차

<스택, 큐, 그리고 스택을 구현해 보자.>

CAPS

Q. 자료구조를 배워야 알고리즘을 하나요?

A. NO!

Hello World 쳐 보셨죠?

여러분들은 이미 알고리즘을 한 겁니다. 너무 무거운 마음으로 하지는 맙시다.

특정 알고리즘에 특정 자료구조가 필요하지만, 포켓몬 텍스트리와같이 딱딱한 개념은 아닙니다.

그래도 아주 필수적인 거 다뤄봅시다.

스택과 큐

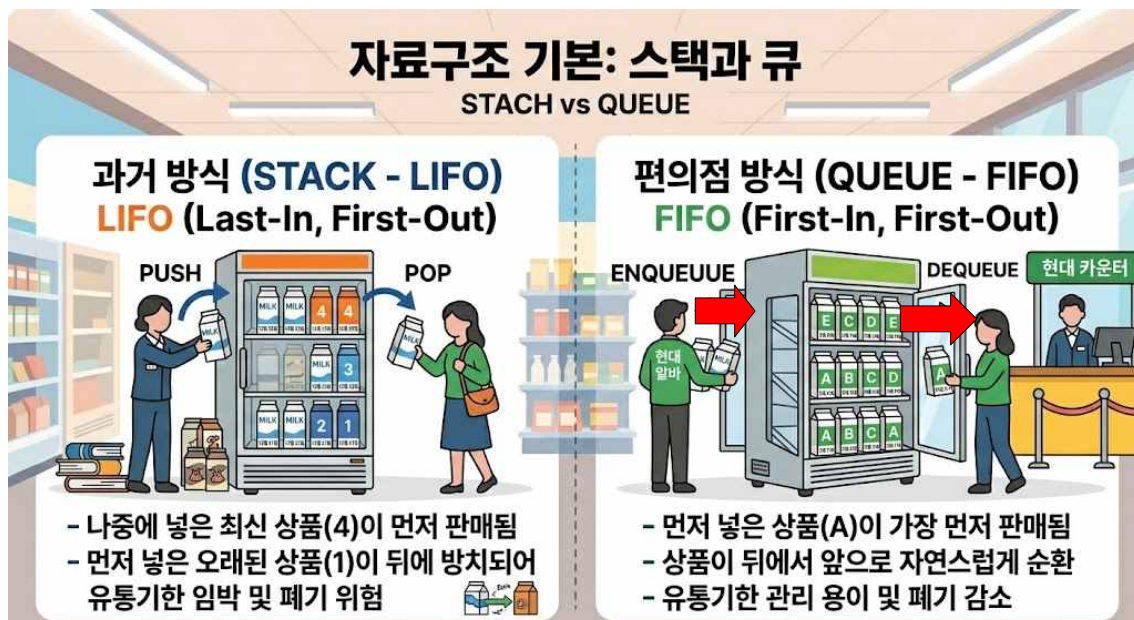
스택은 LIFO, 큐는 FIFO입니다. 굳이 외워야하느냐? 필요없습니다.

1. 스택은 '나중에' 들어간 게 '먼저' 나오구요
2. 큐는 '먼저' 들어간 게 '나중에' 나옵니다.

간단한 자료구조 설명: 스택과 큐



1. 큐가 좋을 때



2. 스택이 좋을 때

가나

가나다 // 다 추가

가나 // 다 삭제 와 이게 스택이지~ 여러분들이 쓰는 글들이 모두 스택입니다.

스택

1. 정직한 스택

```
#include<iostream>
#include<stack> // stack을 잘 include 해 주세요
using namespace std;
stack<int> s;           // {}
int main(void) {
    s.push(1);           // {1}
    s.push(5);           // {1,5}
    s.push(4);           // {1,5,4}
    s.pop();             // {1,5}
    /*
    스택은 원칙적으로 맨 위(오른쪽)의 원소만 볼 수 있습니다.
    */
    cout << s.top() << "\n"; // 5
    cout << s.size() << "\n"; // 2
    cout << s.empty() << "\n"; // 비워져 있으면 1, 차 있으면 0!
}
```

2. 근데 사실

```
#include<iostream>
#include<vector> // vector의 방식이 stack과 비슷하지 않나요?
using namespace std;
vector<int> v;           // {}
int main(void) {
    v.push_back(1);       // {1}
    v.push_back(5);       // {1,5}
    v.push_back(4);       // {1,5,4}
    v.pop_back();         // {1,5}

    cout << v.back() << "\n"; // 5
    cout << v.size() << "\n"; // 2
    cout << v.empty() << "\n"; // 비워져 있으면 1, 차 있으면 0!
}
//어짜피 vector을 많이 쓰는데, stack을 굳이 안 쓰고 vector를 그대로 쓰기도 합니다.
```

큐

```
#include<iostream>
#include<queue> // queue를 잘 include 해 주세요
using namespace std;
queue<int> q; // {}
int main(void) {
    q.push(1); // {1}
    q.push(5); // {1,5}
    q.push(4); // {1,5,4}
    q.pop(); // {5,4}
    /*
    큐는 원칙적으로 맨 아래(왼쪽)의 원소만 볼 수 있습니다.
    */
    cout << q.front() << "\n"; // 5
    cout << q.size() << "\n"; // 2
    cout << q.empty() << "\n"; // 비워져 있으면 1, 차 있으면 0!
}
```

vector, stack, queue의 선언 방식이 전부 다 비슷하지 않던가요?
향후 배우실 많은 자료구조 STL 들이 선언을 그렇게 합니다. 익숙해지면 좋아요.

덱 (데큐, deque) 라는 것도 있는데요, 생각보다 안 씁니다. 궁금하면 배워보시죠. 그래도 메이저한 자료구조입니다. 그냥 앞뒤로 넣고 뺄 수 있는 겁니다.

```
#include<iostream>
#include<deque>
using namespace std;
deque<int> dq; // {}
int main(void) {
    dq.push_back(1); // {1}
    dq.push_back(2); // {1,2}
    dq.push_back(3); // {1,2,3}
    dq.push_front(100); // {100,1,2,3}

    cout << dq.front() << " " << dq.back() << "\n"; // 100 3

    dq.pop_back(); // {100,1,2}
    dq.pop_front(); // {1,2}
    cout << dq.front() << " " << dq.back() << "\n"; // 1 2
}
```

스택 직접 구현하기

<https://www.acmicpc.net/problem/10828>

0	1	2	3	4	5	6	7

스택을 배열로 구현합니다. 빨간색의 역할은 인덱스를 가르키는데, 어떻게 쓸까요?

1. push(1)

0	1	2	3	4	5	6	7
1							

2. push(3)

0	1	2	3	4	5	6	7
1	3						

3. size()

2를 출력하면 되겠죠? 인덱스의 값을 출력하면 되겠습니다.

4. top()

3을 출력하면 됩니다. s[인덱스-1]을 출력하면 되겠네요.

5. pop()

0	1	2	3	4	5	6	7
1	3						

흠.. 조금 이상합니다. 3을 뺐으니 인덱스가 왼쪽으로 가는 건 알겠는데, 배열에는 아직 3이 있습니다. 왜그럴까요?

향후 스택을 통해 사용할 모든 함수에 대해서 '3'이라는 데이터는 쓰일 일이 없습니다. 실질적으로는 사라지지 않았지만, 논리적으로는 삭제된 상태입니다.

이제 저 문제를 풀어볼까요?

정답코드는 아래에 있지만, 여러분들이 해 봅시다.

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/10828.cpp>

```
#include <iostream>
#include <string>
using namespace std;

int arr[10000]; // 배열이구요
int ind = 0;    // 아까 그 빨간 인덱스입니다.
int main(void) {
    ios::sync_with_stdio(0);
    cin.tie(0);
    int Q; cin >> Q;
    while (Q--) {
        string str; cin >> str;

        if (str == "push") {
            int in; cin >> in;
            arr[ind] = in;
            ind++;
        }

        if (str == "pop") {
            if (ind == 0) cout << "-1\n";
            else {
                cout << arr[ind - 1] << "\n";
                ind--;
            }
        }

        if (str == "size") {
            cout << ind << "\n";
        }

        if (str == "empty") {
            cout << (ind == 0) << "\n"; // 어때요 이 테크닉 많이 쓰죠?
        }

        if (str == "top") {
            if (ind == 0) cout << "-1\n";
            else cout << arr[ind - 1] << "\n";
        }
    }
}
```

스택 문제)

1. 10828 스택

<https://www.acmicpc.net/problem/10828>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/10828.cpp>

이 밑부터는 스택 활용 문제입니다. 유형이 어느 정도 정해져 있지만, 처음 맨땅으로 배우기는 엄청 어려울 겁니다. 인터넷의 풀이를 보셔도 괜찮아요.
그리고 STL을 적극 활용해 봅시다.

2. 10773 제로

<https://www.acmicpc.net/problem/10773>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/10773.cpp>

3. 9012 괄호

<https://www.acmicpc.net/problem/9012>

<https://github.com/sungjoonyoung/BOJ/blob/main/00000/9012.cpp>

4. 17298 오큰수 (이거 풀면 밥 사줌)

<https://www.acmicpc.net/problem/17298>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/17298.cpp>

큐 문제)

1. 10845 큐

<https://www.acmicpc.net/problem/10845>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/10845.cpp>

이 밑부터는 큐 활용 문제입니다. 유형이 어느 정도 정해져 있지만, 처음 맨땅으로 배우기는 엄청 어려울 겁니다. 인터넷의 풀이를 보셔도 괜찮아요.
그리고 STL을 적극 활용해 봅시다.

2. 2164 카드2

<https://www.acmicpc.net/problem/2164>

<https://github.com/sungjoonyoung/BOJ/blob/main/00000/2164.cpp>

3. 11866 요세푸스 문제 0

<https://www.acmicpc.net/problem/11866>

<https://github.com/sungjoonyoung/BOJ/blob/main/10000/11866.cpp>