19

큐, 스택, 데크

큐와 스택, 데크

- 도입
- 구현?
- 활용
- 문제

도입

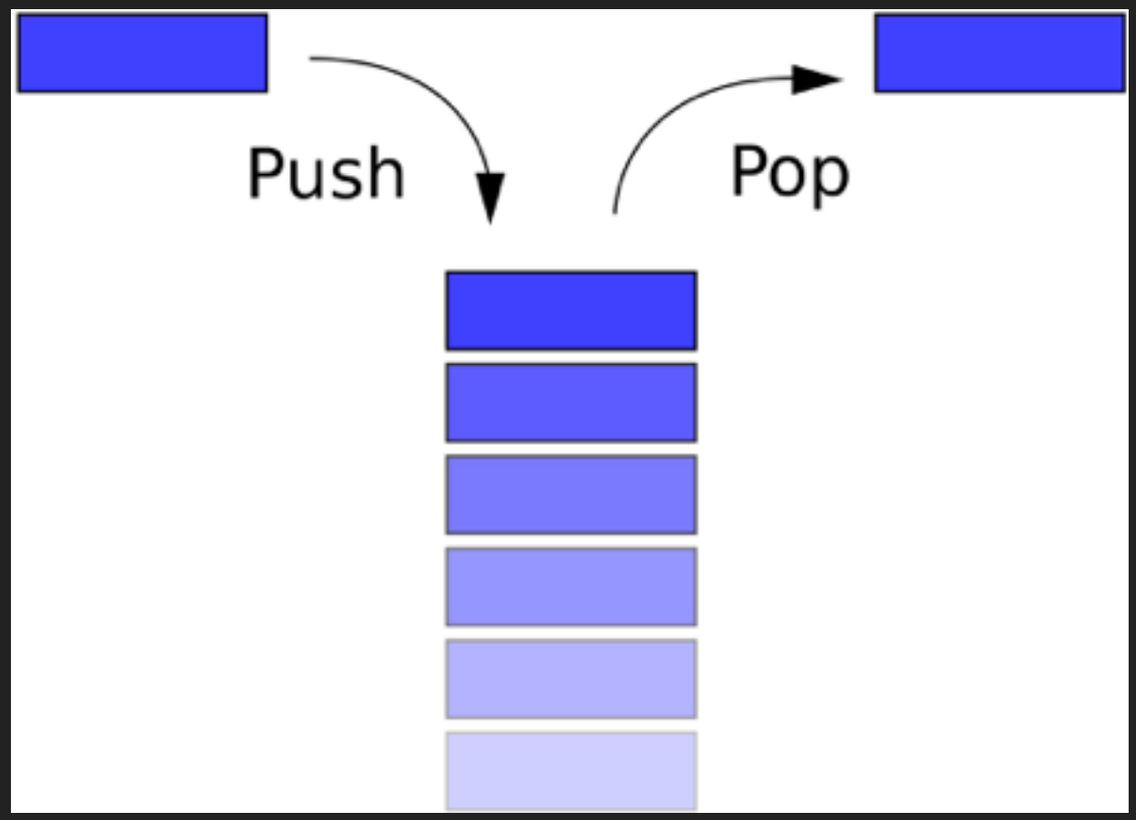
- 특정한 순서로 자료를 넣고 특정한 순서로 자료를 꺼내기
- 다른 알고리즘을 구현하는 도구 (BFS, DFS, Dijkstra, ...)
- 개발자간의 의사소통을 쉽게함
- 배열, 연결 리스트로 구현

QUEUE

- FIRST IN FIRST OUT (선입선출)
- 일상생활과 밀접한 연관

STACK

- FIRST IN LAST OUT(후입선출)
- function call stack (context)



DEQUE (DOUBLE - ENDED QUEUE)

- Deque = Queue + Stack
- 양쪽 끝에서 자료들을 넣고 뺄 수 있는 자료 구조
- https://www.acmicpc.net/problem/10866

아래 연산 들을 상수 시간(O(1)) 에 할 수 있어야 좋은 구현

- push_front X: 정수 X를 덱의 앞에 넣는다.
- push_back X: 정수 X를 덱의 뒤에 넣는다.
- pop_front: 덱의 가장 앞에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- pop_back: 덱의 가장 뒤에 있는 수를 빼고, 그 수를 출력한다. 만약, 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 덱에 들어있는 정수의 개수를 출력한다.
- empty: 덱이 비어있으면 1을, 아니면 0을 출력한다.
- front: 덱의 가장 앞에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- back: 덱의 가장 뒤에 있는 정수를 출력한다. 만약 덱에 들어있는 정수가 없는 경우에는 -1을 출력한다.

구현(1) - 연결 리스트를 이용한 구현

- 추가, 삭제가 쉬움 (push pop)
- 노드를 할당하고 삭제하는 연산, 포인터를 따라가는 연산(스택의 크기 계산) 등이 시간이 걸림

구현(2) - 동적 배열을 이용한 구현

- 스택 : 간단!
- 큐, 데크 : head, tail, circular buffer 이용
- C 구현 : soen.kr 참조

표준 라이브러리 (STL, JAVA) 의 구현

- 기본 자료구조 이기 때문에 라이브러리로 구현되어 있음
- #include <deque>

```
deque<int> d;
d.push_back(1); print(d); //1
d.push_front(2); print(d); // 2 1
d.push_back(3); print(d); // 2 1 3
d.pop_back(); print(d); // 2 1
d.pop_front(); print(d); //1
```

활용

- 조세퍼스 문제
- 울타리 자르기 문제
- …
- 선형 자료구조 이기 때문에 문제 풀이에 사용하게 되면 보통 O(N) 으로 끝나는 경우가 많음

문제 : 짝이 맞지 않는 괄호

- https://www.acmicpc.net/problem/9012
- https://www.acmicpc.net/problem/2504
  - 와 유사한 문제

외계 신호 분석

- https://www.acmicpc.net/problem/1806
- 와 유사한 문제
- 슬라이딩 윈도!

- 온라인 알고리즘(삽입 정렬) vs 오프라인 알고리즘(선택 정렬)
  - 입력 전체가 필요한지 여부에 따라