/* 데이터 구조 및 알고리즘 */

자료구조 디자인

2017. 3. 16.

/* elice */

지난시간 요약

자료구조를 배우는 이유

나의 목적에 맞게 데이터를 담는

그릇을 디자인하기 위함

지난시간 요약

기본적인 자료구조

변수 (Variable)

리스트 (List)

링크드 리스트 (Linked list)

지난시간 요약

리스트 (List)

장점 - i번째 원소를 빠르게 찾을 수 있다

단점 - 원소의 추가 / 삭제가 까다롭다

링크드 리스트 (Linked list)

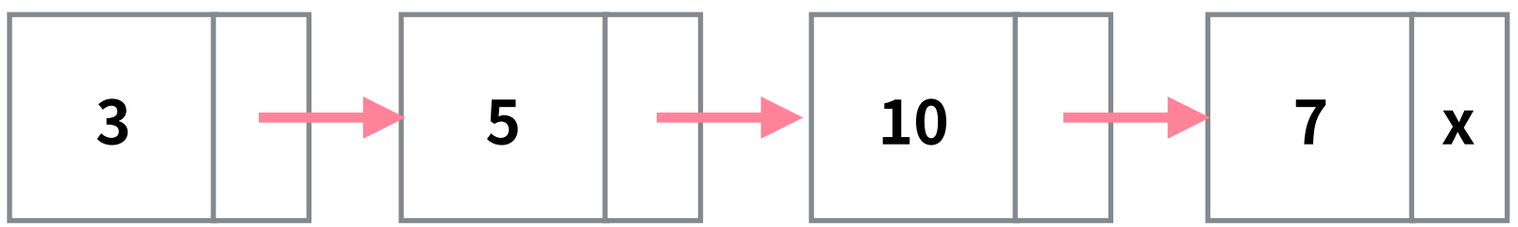
장점 - 원소의 추가 / 삭제를 빠르게 할 수 있다

단점 - i번째 원소를 빠르게 찾기 어렵다

캡슐화 (Encapsulation)

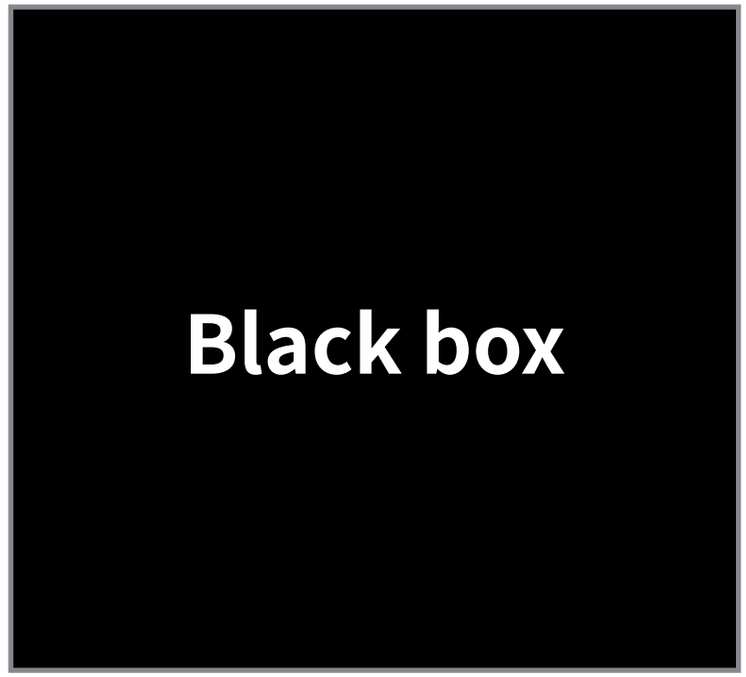
자료구조를 사용하는 사람은

자료구조가 어떻게 동작하는지 알 필요가 없다



4번째 숫자 4를 넣어야지

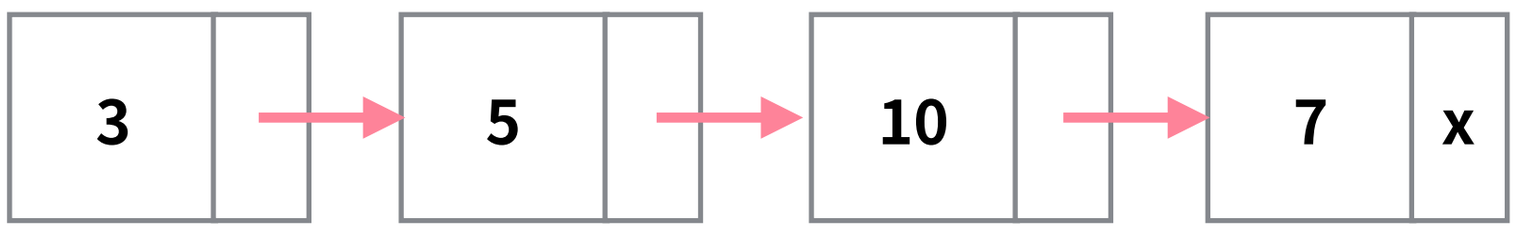
“4번째에 숫자 4좀 넣어줘”



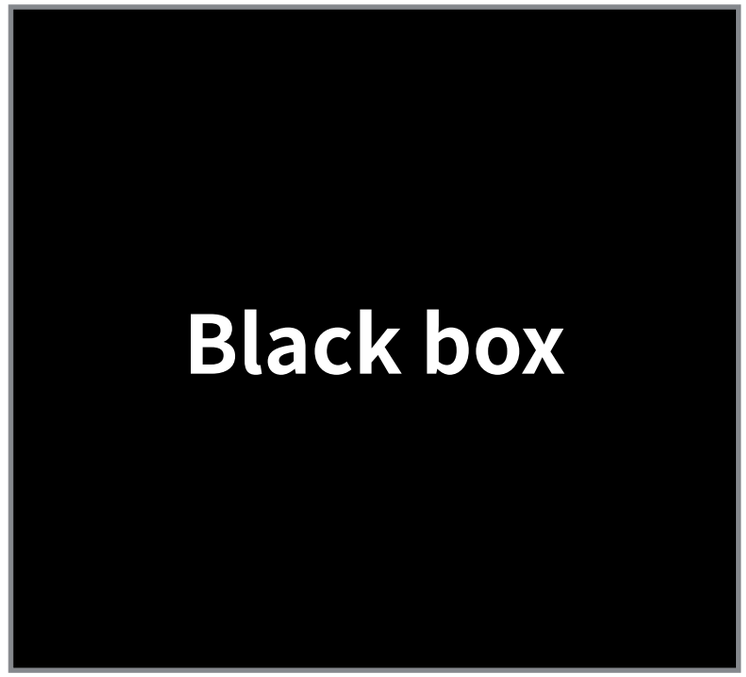
캡슐화 (Encapsulation)

자료구조를 사용하는 사람은

자료구조가 어떻게 동작하는지 알 필요가 없다



4번째 숫자 4를 넣어야지

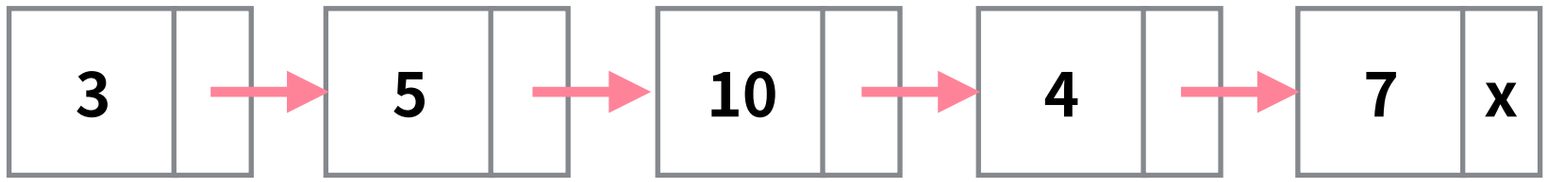


넣었습니다 :)

캡슐화 (Encapsulation)

자료구조를 사용하는 사람은

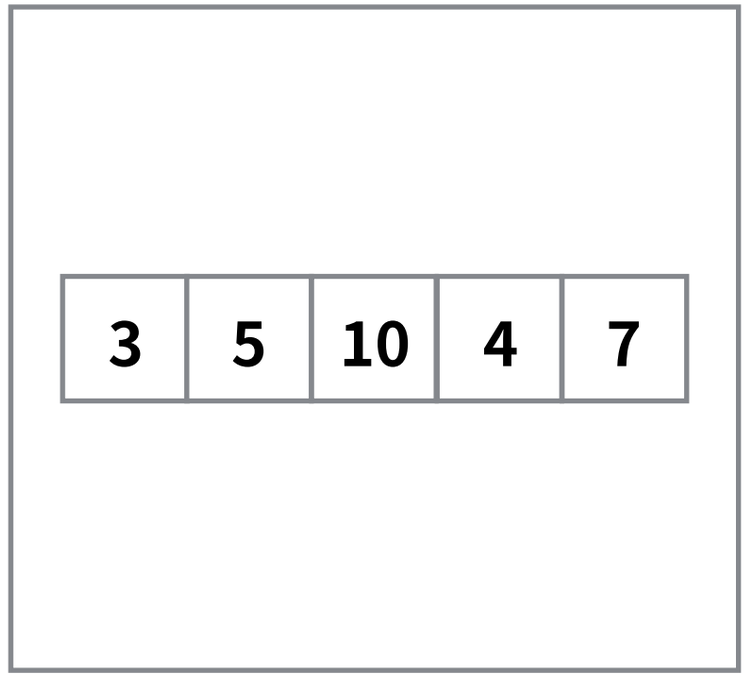
자료구조가 어떻게 동작하는지 알 필요가 없다



그럼 이제 되었겠군

2번째 숫자를 빼자!

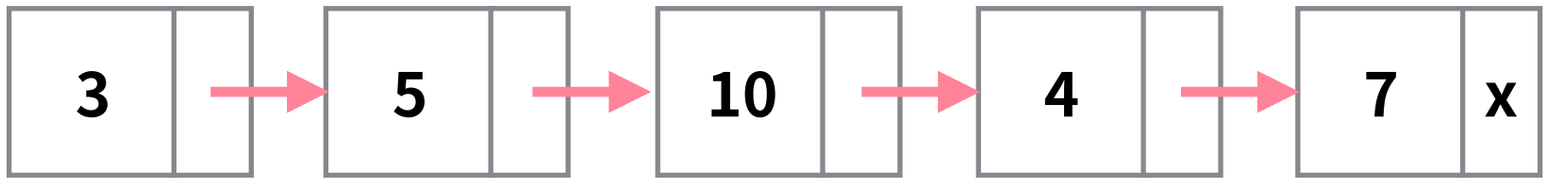
“2번째 숫자 좀 빼줘”

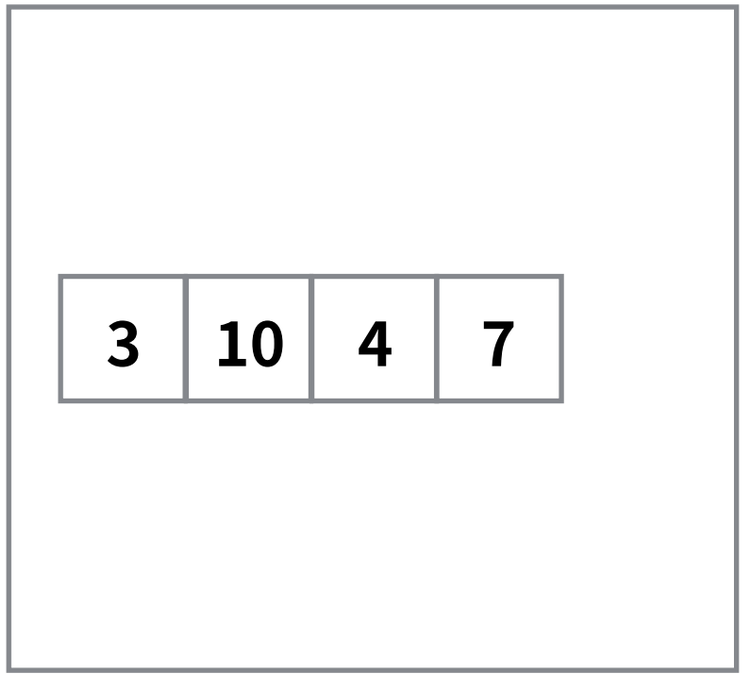


캡슐화 (Encapsulation)

자료구조를 사용하는 사람은

자료구조가 어떻게 동작하는지 알 필요가 없다





그럼 이제 되었겠군

2번째 숫자를 빼자!

뺐습니다 :)

그래서 무엇을 써야하나

리스트를 써야하나

링크드 리스트를 써야하나 ?

그래서 무엇을 써야하나

리스트를 써야하나

링크드 리스트를 써야하나 ?

그때그때 달라요

반드시 숙지할 것

자료구조를 배우는 이유

나의 목적에 맞게 데이터를 담는

그릇을 디자인하기 위함

반드시 숙지할 것

자료구조를 배우는 이유

나의 목적에 맞게 데이터를 담는

그릇을 디자인하기 위함

→ **목적이 먼저, 자료구조는 그 다음!**

반드시 숙지할 것

목적이 먼저, 자료구조는 그 다음!

1. 무슨 자료를 담을지 파악한다.
2. 이 자료를 이용해서 무엇을 할 것인지 파악한다.
3. 목적을 빠르게 달성할 수 있는 자료구조를 디자인한다.

샴푸 통 디자인

+ 디자인이 간단

- 쓰기가 다소 불편

+ 쓰기 좀 더 편함

- 하지만 여전히 불편함

+ 쓰기 간편함

- 많은 양을 얻기 힘듦

+ 우주에서 쓸수있음

- 중력이 있으면 굳이..

샴푸 통 디자인

이용 목적에 따라 그 효율성이 다름

[연습문제] 구슬 넣기

양쪽이 열려있는 파이프에 구슬을 넣을 때,

최종 구슬 배치는 ?

입력의 예

```
3
1 0
2 1
3 0
```

출력의 예

```
3 1 2
```

해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

myList

해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

myList

해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

해법: 리스트 사용

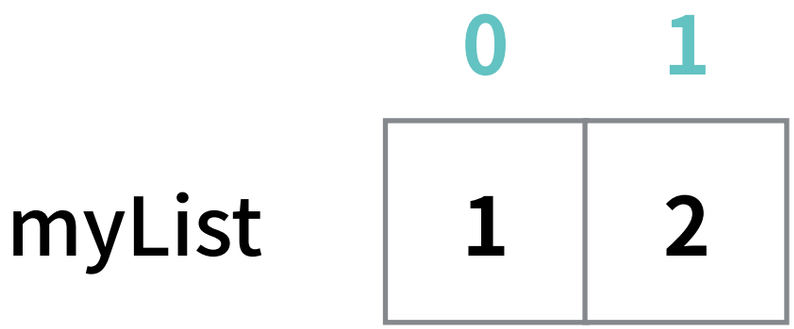
리스트를 파이프라 생각하고 숫자를 삽입하자

오른쪽으로 숫자 2 삽입 !

해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

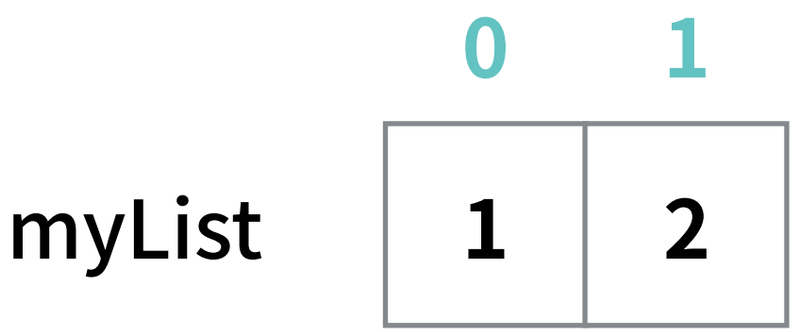
오른쪽으로 숫자 2 삽입 !



해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

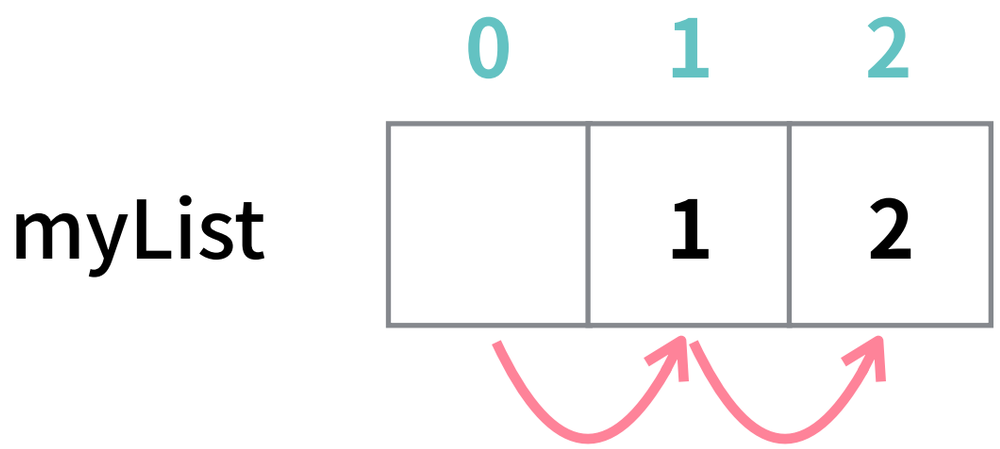
왼쪽으로 숫자 3 삽입 !



해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

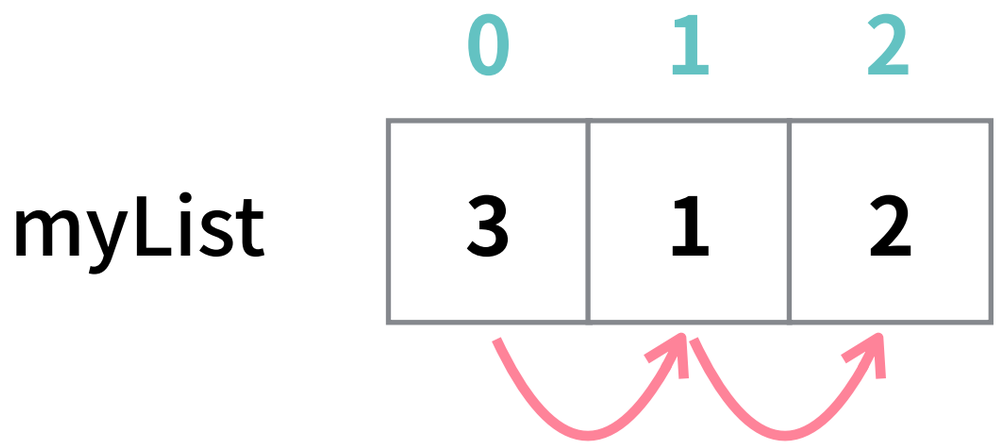
왼쪽으로 숫자 3 삽입 !



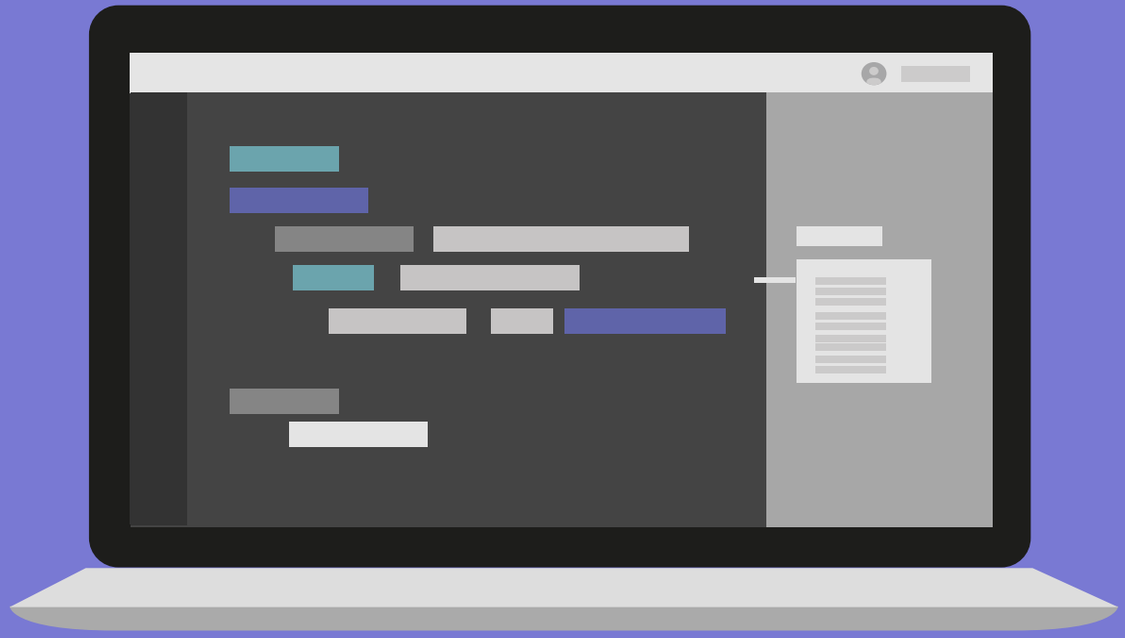
해법: 리스트 사용

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 3 삽입 !



실습



좋은 해법인가?

좋은 해법인가?

기준

- 풀이가 깔끔한가 ?
- **얼마나 빠른가 ?**
- 리소스는 얼마나 잡아먹는가 ?
- 코딩을 하는데는 얼마나 오래 걸리는가 ?

좋은 해법인가?

기준

- ~~풀이가 깔끔한가 ?~~
- **얼마나 빠른가 ?**
- ~~리소스는 얼마나 잡아먹는가 ?~~
- ~~코딩을 하는데는 얼마나 오래 걸리는가 ?~~

시간이 덜 걸리는 코드

수행하는 명령의 수가 적으면 시간이 덜 걸린다

```
sum = 0
for i in range(30)
    sum = sum + 1
```

```
sum = 0
for i in range(300)
    sum = sum + 1
```

시간이 덜 걸리는 코드

수행하는 명령의 수가 적으면 시간이 덜 걸린다

```
sum = 0
for i in range(30)
    sum = sum + 1
```

```
sum = 0
for i in range(300)
    sum = sum + 1
```

거의 10배라고 봐도됨

이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

myList

이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

이 풀이가 수행하는 명령 수

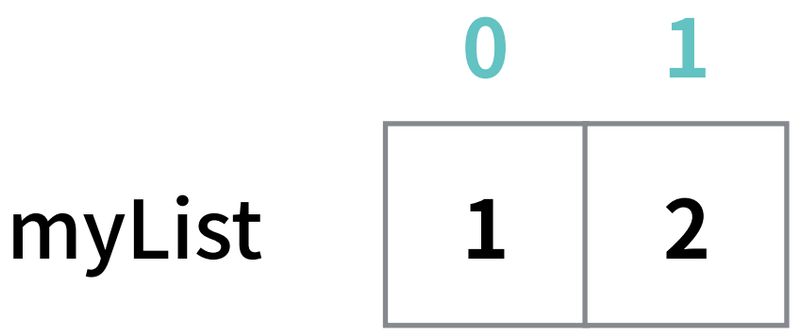
리스트를 파이프라 생각하고 숫자를 삽입하자

오른쪽으로 숫자 2 삽입 !

이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

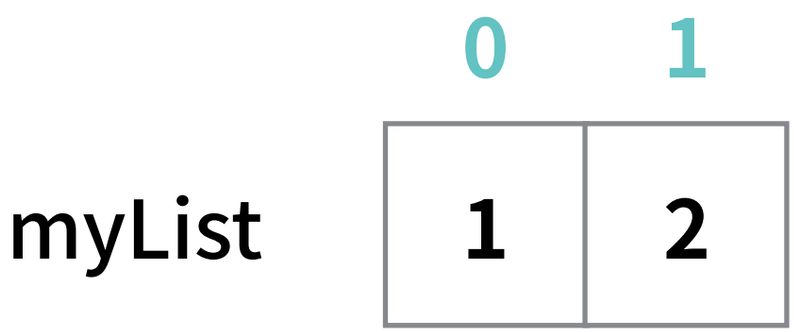
오른쪽으로 숫자 2 삽입 !



이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

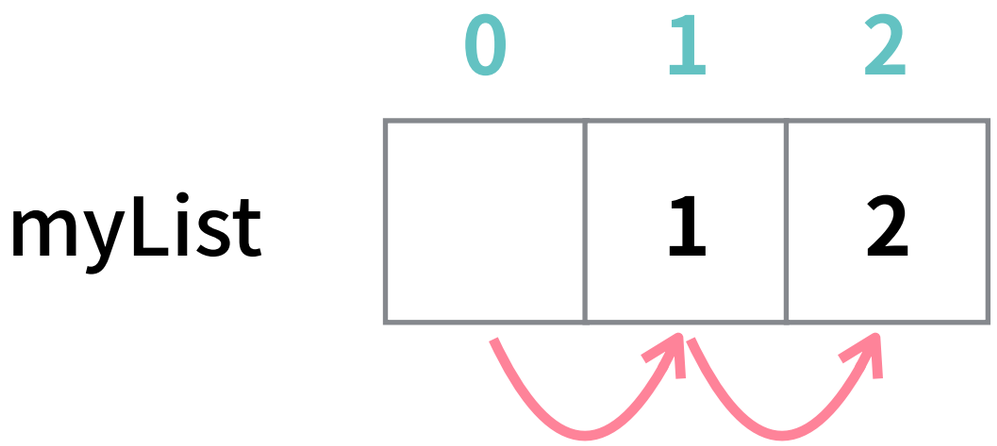
왼쪽으로 숫자 3 삽입 !



이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

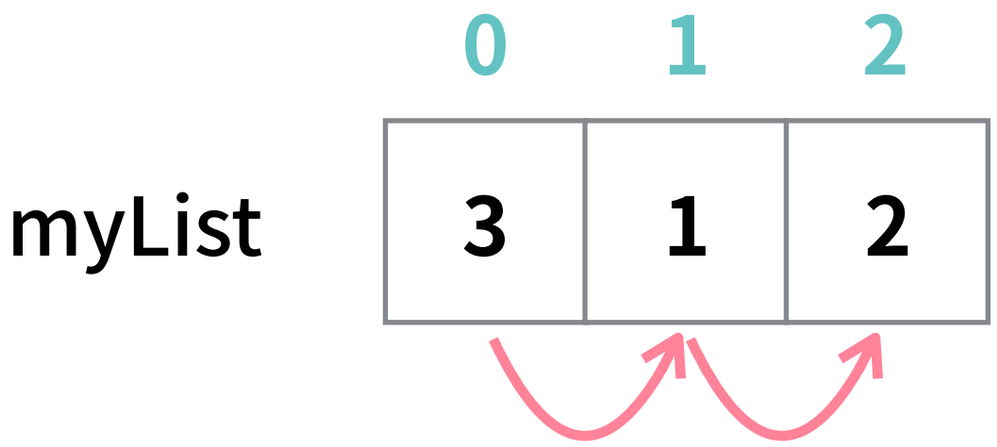
왼쪽으로 숫자 3 삽입 !



이 풀이가 수행하는 명령 수

리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 3 삽입 !



대충 몇개의 명령을 수행할까 ?

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입
- 숫자 하나를 오른쪽으로 삽입

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입 → 숫자 개수만큼
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입 → 숫자 개수만큼
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

운 좋을땐 빠르고, **운 나쁠땐 느리고**...

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입 → 숫자 개수만큼
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

운 좋을땐 빠르고, **운 나쁠땐 느리고**...

- 물론 우리가 항상 운이 나쁘진 않다.
- 하지만 최악의 경우를 대비해서 나쁠건 없다.

최악의 경우에는 어떤가 ?

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

```
1 + 2 + 3 = 6
```

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

```
1 + 2 + 3 = 6
```

만약 숫자가 5개라면

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

```
1 + 2 + 3 = 6
```

만약 숫자가 5개라면

```
1 + 2 + 3 + 4 + 5 = 15
```

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

```
1 + 2 + 3 = 6
```

만약 숫자가 5개라면

```
1 + 2 + 3 + 4 + 5 = 15
```

만약 숫자가 n개라면

최악의 경우에는 어떤가 ?

숫자 n개를 모두 왼쪽으로만 삽입하는 경우

만약 숫자가 3개라면

$$1 + 2 + 3 = 6$$

만약 숫자가 5개라면

$$1 + 2 + 3 + 4 + 5 = 15$$

만약 숫자가 n개라면 $$\frac{n(n+1)}{2}$$

결론

현재 알고리즘은 최악의 경우에 제곱으로 오래걸린다

결론

현재 알고리즘은 최악의 경우에 제곱으로 오래걸린다

더 잘할수는 없나 ?

결론

현재 알고리즘은 최악의 경우에 제곱으로 오래걸린다

더 잘할수는 없나 ?

더 빠른 풀이를 만들 수는 없나 ?

해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

myLinkedList

해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 1 삽입 !

해법: 링크드 리스트 사용

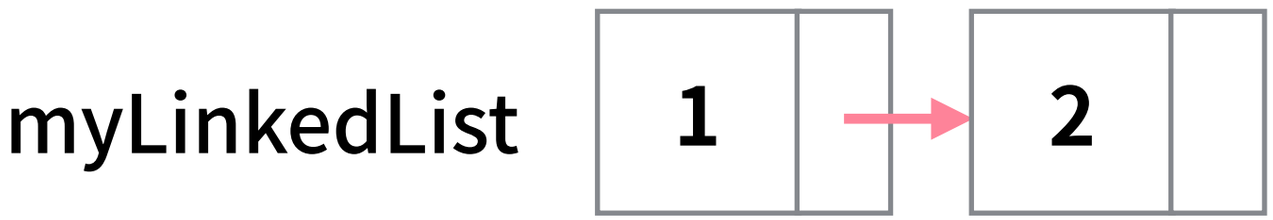
링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

오른쪽으로 숫자 2 삽입 !

해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

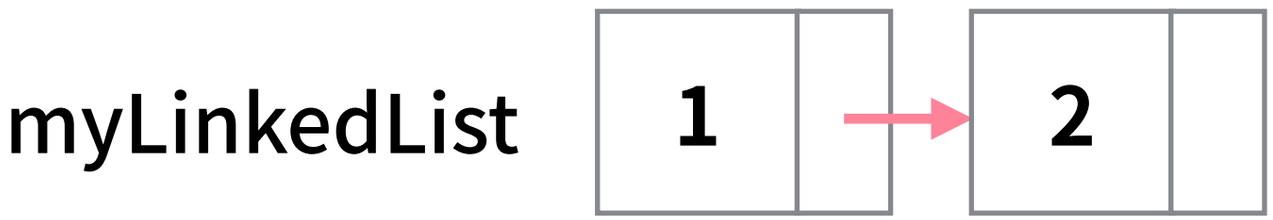
오른쪽으로 숫자 2 삽입 !



해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

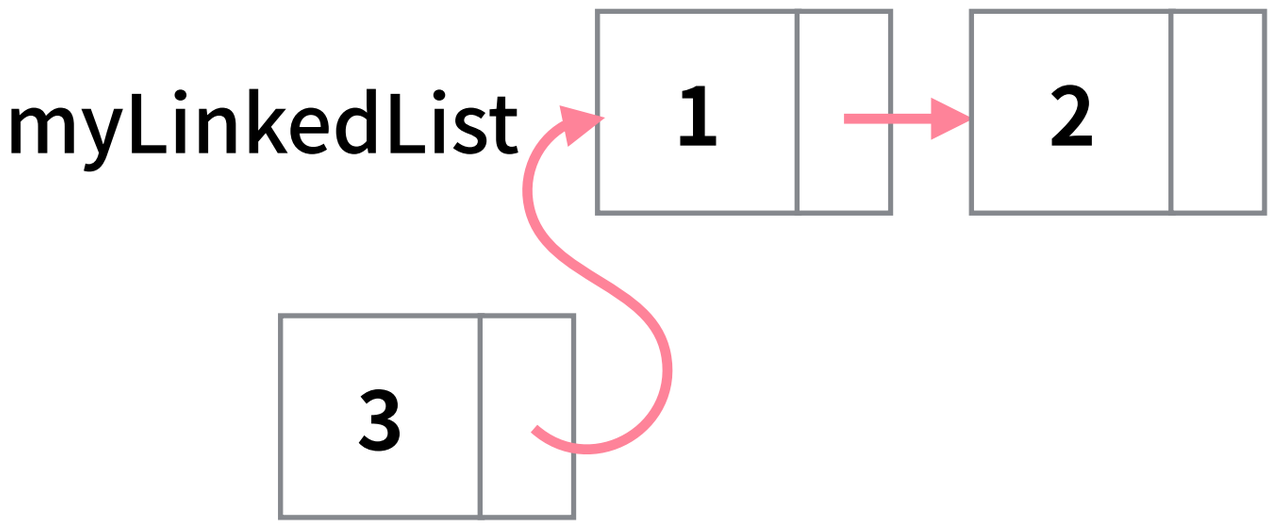
왼쪽으로 숫자 3 삽입 !



해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

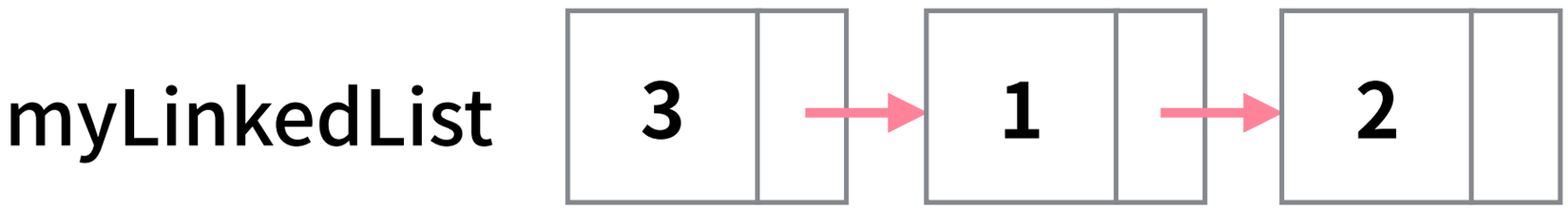
왼쪽으로 숫자 3 삽입 !



해법: 링크드 리스트 사용

링크드 리스트를 파이프라 생각하고 숫자를 삽입하자

왼쪽으로 숫자 3 삽입 !



대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입
- 숫자 하나를 오른쪽으로 삽입

대충 몇개의 명령을 수행할까 ?

우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입 → 1번의 명령
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

대충 몇개의 명령을 수행할까 ?

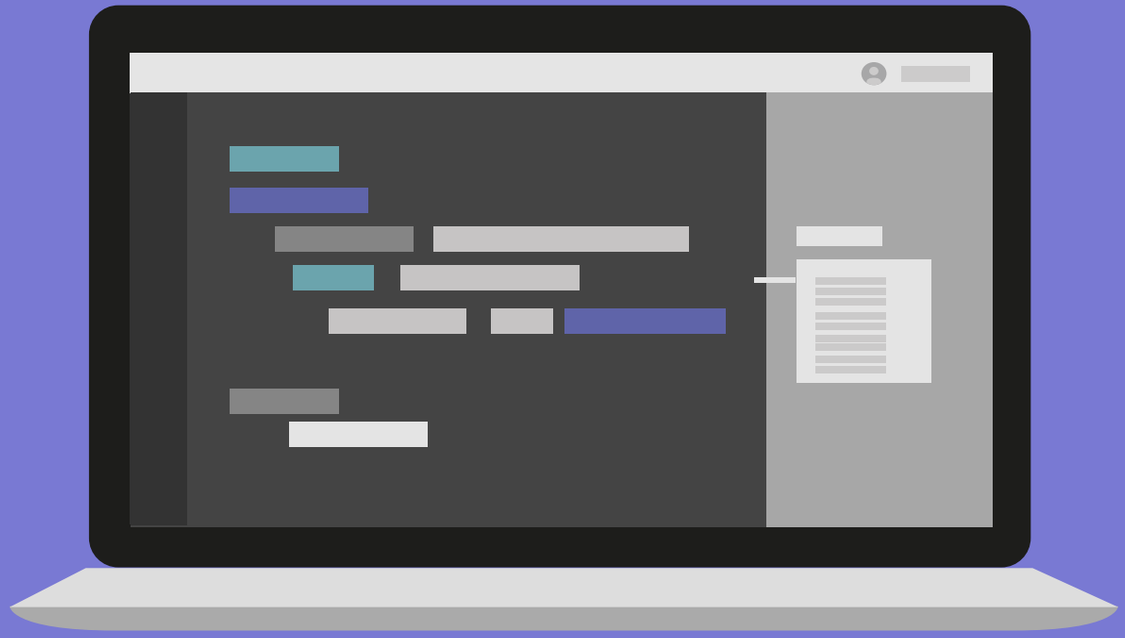
우리 알고리즘이 무슨 일을 하는가 ?

- 숫자 하나를 왼쪽으로 삽입 → 1번의 명령
- 숫자 하나를 오른쪽으로 삽입 → 1번의 명령

운 좋을때 빠르고, **운 나쁠때 빠르고**

- 최악의 경우라 하더라도 1번의 명령만 필요하다

실습



수행 시간 비교

링크드 리스트로 구현한 것이

리스트로 구현한 것보다 **빠르다**

연산을 덜 하기 때문

따라서 이 문제를 풀 때에는

링크드 리스트를 쓰는 것이 더 좋다

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

연산 횟수를 계산

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

연산 횟수를 계산

코딩하기 전에 파악할 수 있음

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

연산 횟수를 계산

코딩하기 전에 파악할 수 있음

목적 달성을 위한 연산 횟수를 줄이는

자료구조를 택하자

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

연산 횟수를 계산

코딩하기 전에 파악할 수 있음

목적 달성을 위한 연산 횟수를 줄이는

자료구조를 택하자

자료구조가 실제로 어떻게 구현되어 있는지를 알아야 함

어느 자료구조를 써야하나 ?

이 문제에서는 링크드 리스트가 더 빨랐음

연산 횟수를 계산

코딩하기 전에 파악할 수 있음

목적 달성을 위한 연산 횟수를 줄이는

자료구조를 택하자

자료구조가 실제로 어떻게 구현되어 있는지를 알아야 함

라이브러리가 어떻게 구현되어 있는지를 모르면 분석할 수 없음

어느 자료구조를 써야하나 ?

연산 횟수를 줄이면 장땡인가 ?

꼭 그렇지는 않지만,

적어도 이번 과정에서는 Yes

감사합니다!

신현규

E-mail : hyungyu.sh@kaist.ac.kr

Kakao : yougatup

/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

blog.naver.com/elicer