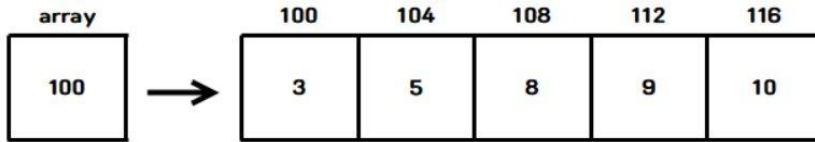


배열의 특징

배열은 같은 타입의 변수들로 이루어진 집합으로 메모리의 연속공간에 값이 채워져 있는 형태의 자료구조이다.



장점:

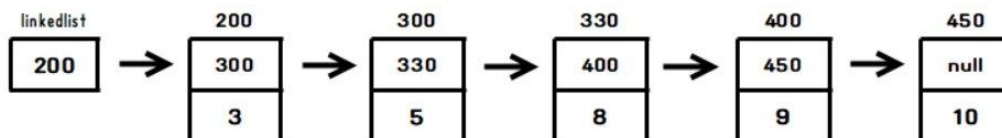
1. 검색 성능이 좋다. 인덱스를 사용하여 원소에 바로 접근할 수 있다.

단점:

1. 초기 사이즈만큼 메모리의 연속공간이 필요하므로 작은 빈 공간은 버려지는 경우가 있어 메모리 활용에 비효율적이다.
2. 값의 삽입과 삭제에서 비효율적이다. 데이터의 중간 지점에서 자료의 삽입, 삭제가 일어날 경우 다음 항목의 모든 값을 이동시켜야만 합니다.

연결리스트 특징

값과 주소를 묶은 노드를 주소로 연결한 자료구조이다.



장점:

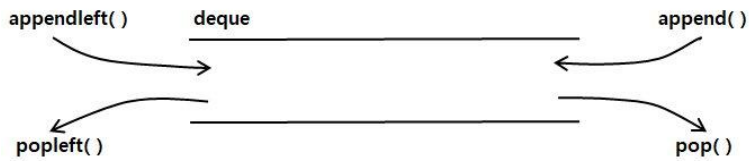
1. 주소로 연결되어 있어 값을 삽입하거나 삭제하는 연산의 속도가 빠르다.
2. 선언할 때 크기를 별도로 지정하지 않고 주소로 계속 연결해 나가며, 연속된 공간이 필요하지 않아 빈 공간을 활용할 수 있어 메모리 활용이 효율적이다.

단점:

1. 리스트 원소로 바로 접근이 불가능하다. Head주소부터 차례대로 순차접근을 해야 한다.

deque 자료구조

아래 그림과 같이 자료구조의 왼쪽 부분과 오른쪽 부분 즉 양쪽 모두에서 자료의 삽입과 삭제가 가능한 자료구조이다.



1. append() : deque의 오른쪽 부분에 자료 추가

```
from collections import deque
dq = deque()
dq.append(1)
dq.append(2)
dq.append(3)
print(dq)          #출력 [1, 2, 3]
```

2. appendleft() : deque의 왼쪽 부분에 자료 추가

```
from collections import deque
dq = deque()
dq.appendleft(1)
dq.appendleft(2)
dq.appendleft(3)
print(dq)          #출력 [3, 2, 1]
```

3. popleft() : deque의 맨 왼쪽 자료 제거

```
from collections import deque
dq = deque()
dq.append(1)
dq.append(2)
dq.append(3)
dq.popleft()
print(dq)          #출력 [2, 3]
```

4. pop() : deque의 맨 오른쪽 자료 제거

최솟값의 위치

수열의 원소에서 가장 작은 값을 찾고 싶습니다.

매개변수 `nums`에 길이가 `n`인 수열이 주어지면 수열의 원소중에서 가장 작은 값을 찾아 그 값의 `nums` 배열의 인덱스 번호를 반환하는 프로그램을 작성하세요.

배열의 인덱스 번호는 0부터 시작합니다.

입출력 예:

nums	answer
[7, 10, 5, 3, 2, 15, 19]	4
[-12, 12, 30, -15, -5, 3, 9, -11, 14]	3
[17, 11, 5, 8, 23, 29, 19, 12, 25, 16, 2]	10
[7, 5, 12, -9, -12, 22, -30, -35, -21]	7

제한사항:

- `nums`의 길이 $3 \leq n \leq 100,000$
- 배열 `nums`의 원소는 정수입니다. $-1,000,000 \leq \text{nums}[i] \leq 1,000,000$
- 배열 `nums`의 원소는 중복된 값이 존재하지 않습니다.

입력예제 1 설명:

[2, 7, 1, 12, 8, 15, 19]의 최솟값은 1이고 1은 배열의 2번 인덱스 위치에 있습니다.

합격생

코딩테스트 자격증 시험에 합격한 수험생 수를 구하려고 합니다.

매개변수 score에 시험을 친 수험생들의 시험 점수가 주어지고, 매개변수 k에 합격 커드라인 점수가 주어지면 자격증 시험에 합격한 수험생의 수를 구해 반환하는 프로그램을 작성하세요.

입출력 예:

score	k	answer
[60, 50, 80, 90, 55, 70, 65, 45]	60	5
[10, 20, 30, 40, 50]	60	0
[50, 65, 75, 87, 90, 55, 78, 93, 100]	70	6
[99, 30, 50, 55, 68, 70, 90, 100]	80	3

제한사항:

- score의 길이 $3 \leq n \leq 10,000$
- $0 \leq \text{score}[i] \leq 100$
- $50 \leq k \leq 90$

연속된 '1'의 길이

매개변수 nums에 0과 1로된 수열이 주어지면 1이 연속된 부분수열 중 가장 긴 부분수열의 길이를 반환하는 프로그램을 작성하세요.

입출력 예:

nums	answer
[1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1]	5
[0, 0, 1, 0, 1, 0, 0]	1
[1, 1, 1, 1, 1]	5
[1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1]	3

제한사항:

- nums의 길이 $3 \leq n \leq 100,000$

수열의 회전

정수 수열의 원소를 회전하고 싶습니다.

매개변수 `nums`에 길이가 `n`인 수열이 주어지고, 매개변수 `k`에 뒤로 이동시키고 싶은 원소의 개수가 주어지면 `nums`의 원소 중 앞 원소 `k`개를 수열의 뒤쪽으로 이동하고 난 후의 수열을 반환하는 프로그램을 작성하세요.

입출력 예:

nums	k	answer
[3, 7, 1, 5, 9, 2, 8]	3	[5, 9, 2, 8, 3, 7, 1]
[2, 12, 2, 1, 3, 3, 9]	2	[2, 1, 3, 3, 9, 2, 12]
[1, 2, 5, 4, 6, 7, 9]	6	[9, 1, 2, 5, 4, 6, 7]
[1, 3, 6, 8, 14, 2, 1, 7]	5	[2, 1, 7, 1, 3, 6, 8, 14]

제한사항:

- `nums`의 길이 $3 \leq n \leq 200,000$
- 배열 `nums`의 원소는 정수입니다. $-10,000 \leq \text{nums}[i] \leq 10,000$
- $0 \leq k \leq \text{nums의 길이}$

중복 제거

오름차순으로 정렬된 수열이 주어지면 중복된 값을 제거하고 유일값으로 구성된 내림차순 수열을 만들고 싶습니다.

매개변수 `nums`에 길이가 `n`인 수열이 주어지면 중복된 값을 제거하고 유일값만으로 구성된 내림차순 수열을 배열에 담아 반환하는 프로그램을 작성하세요.

입출력 예:

nums	answer
[0, 1, 1, 1, 2, 2, 2, 3]	[3, 2, 1, 0]
[1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 5]	[5, 4, 3, 2, 1]
[0, 0, 0, 3, 3, 3, 5, 7, 7, 7]	[7, 5, 3, 0]
[1, 2, 3, 4, 5, 6, 7, 7, 7, 8, 9]	[9, 8, 7, 6, 5, 4, 3, 2, 1]

제한사항:

- `nums`의 길이 $3 \leq n \leq 200,000$
- 배열 `nums`의 원소는 정수입니다. $-10,000 \leq \text{nums}[i] \leq 10,000$

두 수의 합

정수 수열 안에서 수열의 원소 두 개의 합이 target값이 되는 경우를 찾고 싶습니다.

매개변수 nums에 길이가 n인 수열이 주어지고, 매개변수 target에 자연수 값이 주어지면 이 수열안에서 두 개의 원소의 합이 정수 target값이 되는 두 원소를 구해 배열에 오름차순으로 담아 반환합니다.

두 개의 원소의 합이 target값이 되는 경우는 오직 한가지 뿐인 입력만 주어집니다. 한 원소를 두 번 더하는 것은 안됩니다. nums의 각 원소는 유일값입니다.

답이 없을 경우 [0, 0]을 반환합니다.

입출력 예:

nums	target	answer
[7, 3, 2, 13, 9, 15, 8, 11]	12	[3, 9]
[21, 12, 30, 15, 6, 2, 9, 19, 14]	24	[9, 15]
[12, 18, 5, 8, 21, 27, 22, 25, 16, 2]	28	[12, 16]
[11, 17, 6, 8, 21, 9, 19, 12, 25, 16, 2]	26	[9, 17]
[7, 5, 12, -9, -12, 22, -30, -35, -21]	-14	[-21, 7]
[7, 5, 12, 20]	15	[0, 0]

제한사항:

- nums의 길이 $3 \leq n \leq 10,000$
- 배열 nums의 원소는 정수입니다. $-10,000 \leq \text{nums}[i] \leq 10,000$
- $-20,000 \leq \text{target} \leq 20,000$