

파이썬 sort 함수

1. 오름차순 정렬 : `nums.sort()`
`nums` 배열(리스트) 자체를 정렬하고, 반환값은 `None`이다.

2. 내림차순 정렬 : `nums.sort(reverse = True)`

3. 좌표정렬하기

1) `[x, y]`에서 `x`값에 의한 오름차순 정렬

```
def solution(nums):
    nums.sort(key = lambda v : (v[0]))
    return nums;
print(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

2) `[x, y]`에서 `y`값에 의한 오름차순 정렬

```
def solution(nums):
    nums.sort(key = lambda v : (v[1]))
    return nums;
print(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

3) `[x, y]`에서 `x`값에 의한 내림차순 정렬

```
def solution(nums):
    nums.sort(key = lambda v : (-v[0]))
    return nums;
print(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

4) `[x, y]`에서 `y`값에 의한 오름차순을 하되 `y`값이 같은 경우는 `x`값에 따라 오름차순한다.

```
def solution(nums):
    nums.sort(key = lambda v : (v[1], v[0]))
    return nums;
print(solution([[2, 3], [1, 4], [3, 1], [1, 1]]));
```

파이썬 sorted 함수

1. 오름차순 정렬 : sorted(nums)

nums 배열 자체는 바꾸지 않고 정렬된 새로운 배열을 반환한다.

인자로써 배열, 튜플, 문자열 등 순회가 가능한(iterable) 객체는 모두 가능하다.

```
nums = [3, 2, 5, 1, 4]
print(sorted(nums)) #[1, 2, 3, 4, 5]
print(nums) #[3, 2, 5, 1, 4]
```

```
print(sorted('study')) #['d', 's', 't', 'u', 'y']
print(''.join(sorted('study'))) #'dituy'
```

2. 내림차순 정렬 : sorted(nums, reverse=True)

```
nums = [3, 2, 5, 1, 4]
print(sorted(nums, reverse=True)) #[5, 4, 3, 2, 1]
print(nums) #[3, 2, 5, 1, 4]
```

사탕 종류

현수는 사탕을 좋아합니다. 현수에게 사탕이 n 개 있습니다.

현수 엄마는 현수가 요즘 너무 사탕을 많이 먹어 건강에 좋지 않다고 생각해 현수에게 가지고 있는 사탕의 절반($n/2$)개만 먹으라고 했습니다. n 은 항상 짝수입니다.

매개변수 `nums`에 현수가 가지고 있는 n 개의 사탕의 종류 정보가 주어진다면 현수가 $n/2$ 개의 사탕을 먹는다면 최대 몇 종류의 사탕을 먹을 수 있는지를 반환하는 프로그램을 작성하세요.

입출력 예:

nums	answer
[2, 1, 1, 3, 2, 3, 1, 2]	3
[1, 3, 5, 7, 2, 3, 7, 5, 3, 2, 5, 7, 9, 12]	7
[5, 5, 5, 5, 5, 5]	1
[12, 23, 11, 3, 5, 23, 23, 23, 23, 23, 23]	5
[100, 200, 300, 400, 500, 600, 700, 800]	4

제한사항:

- `nums`의 길이는 100,000을 넘지 않습니다.
- `nums[i]`는 i 번째 사탕의 종류를 의미합니다. `nums[i]`값이 같으면 같은 종류의 사탕입니다.
- $1 \leq \text{nums}[i] \leq 1,000$

두 수의 차

매개변수 `nums`에 수열이 주어지면 수열의 원소 중 두 수의 차가 가장 작은 쌍을 찾아 반환하는 프로그램을 작성하세요.

수열의 원소는 유일값들로 이루어져 있습니다. 두 수의 차가 가장 작은 쌍이 여러개면 모든 쌍을 배열에 담아 반환합니다. 배열에 담는 순서는 상관없습니다. 단 두 수는 오름차순 정렬된 쌍으로 표현합니다. 정확성, 효율성테스트를 합니다.

입출력 예:

nums	answer
[3, 8, 1, 5, 12]	[[1, 3], [3, 5]]
[2, 1, 3, 5, 4]	[[1, 2], [2, 3], [3, 4], [4, 5]]
[5, 10, 15, 20, 25, 11]	[[10, 11]]
[2, 4, 3, 1, 5, 7, 8, 12, 13, 15, 23]	[[2, 3], [1, 2], [3, 4], [4, 5], [7, 8], [12, 13]]
[100, 200, 300, 400, 120, 130, 135, 132, 121]	[[120, 121]]

제한사항:

- `nums`의 길이는 100,000을 넘지 않습니다.
- $1 \leq \text{nums}[i] \leq 1,000$

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 $3 \leq n \leq 200,000$
- 배열 nums의 원소는 정수입니다. $-10,000 \leq \text{nums}[i] \leq 10,000$
- $-20,000 \leq \text{target} \leq 20,000$