

선택정렬

선택정렬은 첫 번째 자료 위치에 첫 번째 자료부터 마지막 자료까지에서 가장 작은 값을 위치시키고, 두 번째 자료 위치에 두 번째 자료부터 마지막 자료까지에서 가장 작은 값을 위치시키는 이 과정을 계속 반복해서 오름차순 정렬을 시키는 정렬방법입니다.

만약 [5, 4, 2, 1, 3] 배열을 선택정렬로 오름차순 하는 과정은

회전	배열(회전결과)	index: 최솟값
0	[5, 4, 2, 1, 3]	3: 1
1	[1, 4, 2, 5, 3]	2: 2
2	[1, 2, 4, 5, 3]	4: 3
3	[1, 2, 3, 5, 4]	4: 4
4	[1, 2, 3, 4, 5]	

매개 변수 nums에 배열이 주어지면 nums를 선택정렬 방법으로 오름차순 정렬하세요.

입출력 예

nums	result
[2, 8, 3, 6, 1, 7, 5, 9]	[1, 2, 3, 5, 6, 7, 8, 9]
[7, 6, 10, 9, 8]	[6, 7, 8, 9, 10]

제한사항

- nums의 길이는 1,000을 넘지 않습니다.

버블정렬

버블정렬은 인접한 두 자료를 비교하여 크기가 순서대로 되어 있지 않으면 두 자료를 교환하는 방법으로 정렬하는 알고리즘입니다. 1회전이 끝나고 나면 가장 큰 자료가 맨 뒤에 위치하게 됩니다.(오름차순)

만약 [5, 4, 2, 1, 3] 배열을 선택정렬로 오름차순 하는 과정은

회전	배열(회전결과)
0	[5, 4, 2, 1, 3]
1	[4, 2, 1, 3, 5]
2	[2, 1, 3, 4, 5]
3	[1, 2, 3, 4, 5]
4	[1, 2, 3, 4, 5]

매개 변수 nums에 배열이 주어지면 nums를 버블정렬 방법으로 오름차순 정렬하세요.

입출력 예

nums	result
[2, 8, 3, 6, 1, 7, 5, 9]	[1, 2, 3, 5, 6, 7, 8, 9]
[7, 6, 10, 9, 8]	[6, 7, 8, 9, 10]

제한사항

- nums의 길이는 1,000을 넘지 않습니다.

삽입정렬

삽입정렬은 배열의 모든 요소를 자신의 앞 자료까지 차례대로 이미 정렬된 배열 부분과 비교하여, 자신의 위치를 찾아 삽입함으로써 정렬을 완성하는 알고리즘이다.

만약 [5, 4, 2, 1, 3] 배열을 선택정렬로 오름차순 하는 과정은

회전	배열(회전결과)
0	[5, 4, 2, 1, 3]
1	[4, 5, 2, 1, 3]
2	[2, 4, 5, 1, 3]
3	[1, 2, 4, 5, 3]
4	[1, 2, 3, 4, 5]

매개 변수 nums에 배열이 주어지면 nums를 삽입정렬 방법으로 오름차순 정렬하세요.

입출력 예

nums	result
[2, 8, 3, 6, 1, 7, 5, 9]	[1, 2, 3, 5, 6, 7, 8, 9]
[7, 6, 10, 9, 8]	[6, 7, 8, 9, 10]

제한사항

- nums의 길이는 1,000을 넘지 않습니다.

교환 횟수

원소값이 서로 다른 n 개의 원소로 이루어진 `nums` 배열이 있습니다.

1. $i=0$ 으로 초기화하고 정렬알고리즘이 시작됩니다.
2. `nums[i]`부터 `nums[n-1]`까지의 최소값을 구합니다. `minIndex`는 최소값이 위치한 인덱스 번호라고 가정합니다.
3. i 와 `minIndex` 값이 다르면 `nums[i]`와 `nums[minIndex]`의 값을 교환합니다.
4. i 값을 1 증가시킵니다.
5. 2~4과정을 반복하면서 오름차순 정렬을 합니다.

회전	배열(회전결과)	minIndex: 최소값
$i = 0$	[5, 4, 2, 1, 3]	3: 1
$i = 1$	[1, 4, 2, 5, 3]	2: 2
$i = 2$	[1, 2, 4, 5, 3]	4: 3
$i = 3$	[1, 2, 3, 5, 4]	4: 4
	[1, 2, 3, 4, 5]	

배열의 각 위치에서 값이 바뀐 횟수를 구해봤습니다.

위치	값 변경 과정	바뀐 횟수
0	5-->1	1
1	4-->2	1
2	2-->4-->3	2
3	1-->5-->4	2
4	3-->4-->5	2

매개변수 `nums`에 서로 다른 값으로 이루어진 배열이 주어지면 각 위치의 바뀐 횟수를 인덱스 번호 순으로 배열에 저장해 반환하는 프로그램을 작성하세요.

입출력 예

nums	result
[5, 4, 2, 1, 3]	[1, 1, 2, 2, 2]
[6, 7, 8, 9, 10]	[0, 0, 0, 0, 0]

제한사항

- `nums`의 길이는 1,000을 넘지 않습니다.

자바스크립트 sort 함수

nums=[2, 5, 3, 1, 4]

1. 오름차순 정렬 : `nums.sort()` : 한 자리 숫자만 정렬이 제대로 됩니다.

2. 내림차순 정렬 : `nums.sort((a, b)=>b-a);`

3. 두 자릿수 이상의 숫자를 오름차순 정렬 : `nums.sort((a, b)=>a-b);`

만약 [2, 5, 3, 1, 4, 10]을 `nums.sort()`로만 하면 [1, 10, 2, 3, 4, 5]로 정렬됩니다.

`sort`함수는 기본값으로 정렬하면 배열 원소를 문자열로 변환한 다음 정렬을 합니다.

4. 좌표정렬하기

1) [x, y]에서 x값에 의한 오름차순 정렬

```
function solution(nums) {
    nums.sort((a, b)=>a[0]-b[0]);
    return nums;
}
console.log(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

2) [x, y]에서 y값에 의한 오름차순 정렬

```
function solution(nums) {
    nums.sort((a, b)=>a[1]-b[1]);
    return nums;
}
console.log(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

3) [x, y]에서 x값에 의한 내림차순 정렬

```
function solution(nums) {
    nums.sort((a, b)=>b[0]-a[0]);
    return nums;
}
console.log(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

4) [x, y]에서 x값에 의한 오름차순을 하되 x값이 같은 경우는 y값에 따라 오름차순한다.

```
function solution(nums) {
    nums.sort((a, b)=>{
        if(a[0]===b[0]) return a[1]-b[1];
        else return a[0]-b[0];
    });
    return nums;
}
console.log(solution([[2, 3], [1, 4], [3, 1], [1, 2]]));
```

카드 가져가기

현수와 영희는 카드게임을 합니다. 테이블에는 n 개의 카드가 있습니다.

n 은 항상 짝수 개수입니다.

n 개의 카드에는 자연수가 적혀 있습니다.

현수와 영희는 한 라운드에 1개의 카드를 가져갑니다. 총 $n/2$ 라운드를 합니다.

$n/2$ 라운드 결과 가져간 카드에 적힌 숫자의 총합이 게임의 점수이고 점수가 큰 사람이 이기는 게임입니다.

현수는 Lady first 정신에 따라 매 라운드 영희가 먼저 카드를 가져가게 합니다.

대신 현수는 k 번의 우선권을 가질 수 있습니다. 우선권을 쓰는 라운드에서는 현수가 먼저 카드를 가져갑니다.

만약 카드에 적힌 숫자가 [7, 8, 5, 9, 3, 1, 3, 1, 1, 9] 이고, $k=2$ 이면

1. 첫 번째 라운드에서는 영희가 9를 가져가면 현수도 9를 가져가므로 우선권을 사용하지 않습니다. 남은 카드는 [7, 8, 5, 3, 1, 3, 1, 1]입니다.

2. 두 번째 라운드에서 영희가 8를 가져가면 현수는 7를 가져갑니다.

남은 카드는 [5, 3, 1, 3, 1, 1]입니다.

3. 세 번째 라운드에서 영희가 5를 가져가고 현수가 3을 가져가야 하지만 현수는 우선권을 써서 현수가 5를 가져가고 영희가 3을 가져갑니다. 남은 카드는 [1, 3, 1, 1]입니다.

4. 네 번째 라운드에서 우선권을 써서 현수가 3, 영희가 1를 가져갑니다. 남은 카드는 [1, 1]

5. 마지막 라운드에서 각자 1를 가져갑니다.

현수가 얻은 카드에 적힌 숫자의 총합은 $9+7+5+3+1 = 25$ 입니다 25는 k 번의 우선권을 사용해서 현수가 얻을 수 있는 최대 점수입니다.

매개변수 `nums`에 카드에 적힌 숫자가 주어지고, k 가 주어지면 현수가 얻을 수 있는 게임의 최대 점수를 반환하는 프로그램을 작성하세요.

입출력 예

nums	k	result
[7, 8, 5, 9, 3, 1, 3, 1, 1, 9]	2	25
[8, 2, 12, 12, 12, 12, 2, 2]	2	34

제한사항

- `nums`의 길이는 200,000을 넘지 않습니다.
- $0 \leq k \leq n/2$
- 현수와 영희는 게임을 이기기 위해 최선을 다 합니다.

이진수 정렬

매개변수 `nums`에 숫자가 주어지면 `nums`의 원소들을 이진수로 변환했을 때 1의 개수가 적은 것부터 많은 것 순으로 정렬하여 반환하는 프로그램을 작성하세요.

만약 `nums = [5, 6, 7, 8, 9]`이고 이 원소들을 이진수로 변환하면

5 --> 101 : 1이 2개

6 --> 110 : 1이 2개

7 --> 111 : 1이 3개

8 --> 1000 : 1이 1개

9 --> 1001 : 1이 2개

이고, 이 수들을 이진수에서 1의 개수에 의해 오름차순 정렬하면 [8, 5, 6, 9, 7]이다.

위에 5, 6, 9는 이진수로 변환했을 때 1의 개수가 2개로 동일하면 십진수가 작은순(오름차순)으로 정렬합니다.

입출력 예

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

제한사항

- `nums`의 길이는 1,000을 넘지 않습니다.
- `nums`의 원소는 10,000을 넘지 않습니다.