두 개 합 (TwoSum)
일일 온도 (Daily Temperature)
서브어레이 최대값 (Maximum Subarray)
그룹 아나그램 (Group Anagrams)
빗물 담기 (Trapping Rain Water)
누락 범위 (Missing Ranges)
나선형 매트릭스 (SpiralMatrix)

Array (시험에 나오는거 위주로)

1. 개념

1) Math.max(max, a) => max 구하기
2) Map+ Array => two sum 문제
3) array+ stack => Daily Temperature
4) sum = sum+ nums[] => sum += nums[]

두개 합(TwoSum)

설명

정수의 배열 nums와 정수 target이 주어집니다.
배열 nums에서 두 숫자의 값을 더하여 target값과 동일할 경우 두 숫자의 인덱스를 리턴합니다

각 입력에 정확히 하나의 솔루션 이 있다고 가정하며, 동일한 요소를 두 번 사용할 수 없습니다 .

Note) 시간복잡도O(n)으로 수행하세요

입출력

Input: nums = [2, 8, 11, 14], target= 16
Output: [1,4]
Because nums[0]+nums[3] = 2 + 14 = 16
return [1, 4].

Input: nums = [3,2,4], target = 6
Output: [2,3]

문제 Format

```
class Solution {
  public int[] solve(int[] nums, int target) { }
}
```

제한사항

$2 <= nums.length <= 10^4$
$-10^9 <= nums[i] <= 10^9$
$-10^9 <= target <= 10^9$

일일온도(Daily Temperature)

설명

일일 온도를 나타내는 int 배열(temperatures)이 주어집니다.
더 따뜻한 날씨의 날을 얻기 위해 해당날짜 이후에 기다려야하는 날짜의 수를 배열로 리턴하세요 .
더 따뜻한 날이 오지 않는다면 0을 리턴하세요.

입출력

Input: temperatures = [73,74,75,71,69,72,76,73]
Output: [1,1,4,2,1,1,0,0]

Input: temperatures = [30,40,50,60]
Output: [1,1,1,0]

문제 Format

```
class Solution {
 public int[] solve(int[] nums) { }
}
```

제한사항

1 <= temperatures.length <= 10^5
30 <= temperatures[i] <= 100

SubArray 최대값(Maximum Subarray)

설명

정수 배열 nums가 주어지면 합계가 가장 큰 연속 하위 배열 (최소한 하나의 숫자 포함)을 찾아서 합계를 리턴합니다.

Note) 시간복잡도 O(n)으로 구하세요.

입출력

Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: [4,-1,2,1]

Input: nums = [5,4,-1,7,8]
Output: 23

문제 Format

```
class Solution {
 public int solve(int[] nums) { }
}
```

제한사항

$1 <= nums.length <= 3 * 10^4$
$-10^5 <= nums[i] <= 10^5$

시간복잡도/공간복잡도 계산

시간복잡도

1. 대상(Source) : 문제에서 입력받은 파라미터(array 등) (속도)
   Time Complexity : O(N)
   대상 : int[] nums
   이유 : for문 실행.

공간복잡도

2. 대상(Source) : 실제 사용되는 저장 공간을 계산(메모리 사용량)
   예) 프로그램을 실행 및 완료하는데 필요한 저장공간
   Space Complexity : O(1)
   대상 : curMax, allMax
   이유 : 다른 저장공간 사용안함

참고

O(1) : 스택, 큐,Map
O(n) : for문 => 데이터를 한번씩 다 호출하니까 (제일 많음)
O(log N) : sort, prirotiyQueue, binary Search Tree, Tree
O(Klog N) : k번만큼 소팅하는 경우
O(n^2) : 이중for문
O(m*n) : 이중for문인데, n이 다른경우bfs,dfs 류 (예 n=100 인데 m=5인경우)

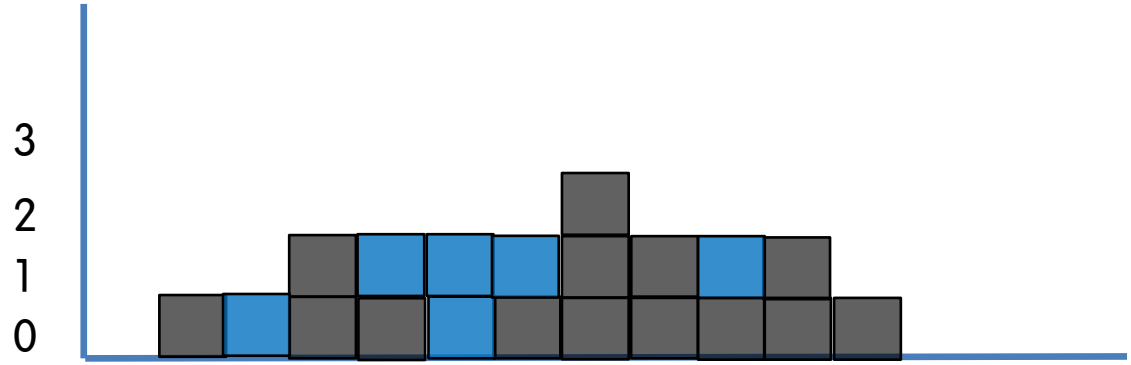
빗물 담기(Trapping Rain Water)

설명

elevation map(양의 정수)은 아래그림에서 검은 네모모양이며 너비가 1이다
비가 내린 후 가둘 수있는 물의 양을 계산합니다.

입출력

Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]
Output: 6
Explanation: elevation map (검은부분) 표시 부분 [0,1,0,2,1,0,1,3,2,1,2,1].
빗물부분 (파란색 부분)은 6개 unit임



누락된 범위(Missing Ranges)

설명

모든 요소를 포함하는 범위[lower, upper]와 정렬된 고유한 정수 배열 nums와 주어집니다.
만약에 x라는 number가 범위[lower, upper]존재하고 nums배열에 없다면
누락된 값으로 간주합니다.
누락된 모든 숫자를 정확히 포함하는 가장 작은 정렬된 범위를 리턴합니다.
표현형식은 아래와 같습니다.

[a,b]목록의 각 범위 는 다음과 같이 출력되어야합니다.
"a->b" if a != b
"a" if a == b

입출력

Input: nums = [2,3,5,50,75], lower = 0, upper = 99
Output: [0->1, 4, 6->49, 51->74, 76->99]
Explanation:
전체 범위가 0-99 입니다.
nums[0]==2입니다. 0,1 빠졌으므로 0->1로표시

나선형매트릭스(SpiralMatrix)

설명

Given a matrix of m x n elements (m rows, n columns),
return all elements of the matrix in spiral order.

입출력

Input:
int[][] matrix = {
{ 1, 2, 3, 4},
{ 5, 6, 7, 8},
{ 9 ,10, 11, 12}
};
출력
[1, 2, 3, 4, 8, 12, 11, 10, 9, 5, 6, 7]

제한사항

m == matrix.length
n == matrix[i].length
1 <= m, n <= 10
-100 <= matrix[i][j] <= 100

문제 Format

```
class Solution {
    public int solve(int[][] matrix) {
    }}
```