
11. Unique Email

12. Longest Substring With At Most Two Distinct

13. Maximum Subarray

14. Find Anagrams Mapping

15. Find All Anagrams

16. SpiralMatrix

17. Group Anagrams

18. Trapping Rain Water

19. Kth Largest Element In An Array

20. Missing Ranges

문제 11) Unique Email Address

Problem

Every email consists of a local name and a domain name, separated by the @ sign.

For example, in `alice@leetcode.com`, `alice` is the local name, and `leetcode.com` is the domain name.

Besides lowercase letters, these emails may contain `'.'`s or `'+'`s.

If you add periods (`'.'`) between some characters in the local name part of an email address, mail sent there will be forwarded to the same address without dots in the local name. For example, `"alice.z@leetcode.com"` and `"alicez@leetcode.com"` forward to the same email address. (Note that this rule does not apply for domain names.)

If you add a plus (`'+'`) in the local name, everything after the first plus sign will be ignored. This allows certain emails to be filtered, for example `m.y+name@email.com` will be forwarded to `my@email.com`. (Again, this rule does not apply for domain names.)

It is possible to use both of these rules at the same time.

Given a list of emails, we send one email to each address in the list. How many different addresses actually receive mails?

Unique Email Address

Problem

Input: [

"test.email+**james**@coding.com",

"test.e.mail+**toto.jane**@coding.com",

"testemail+**tom**@cod.ing.com"]

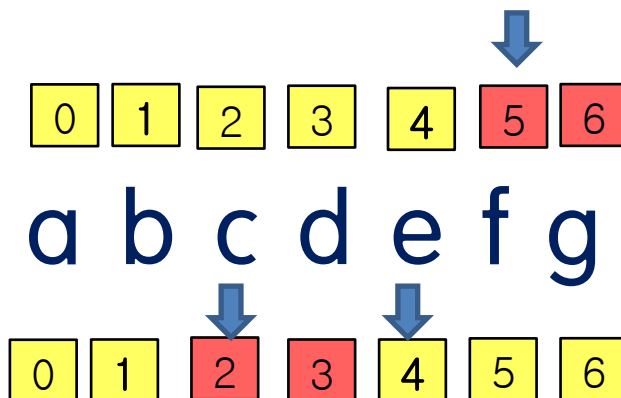
Output: 2

"testemail@coding.com"

"testemail@cod.ing.com"

solution

1. . Continue로 뺀다
2. + break로 뺀다
3. Set<String>



1. a.substring(5); => fg

2. a.substring(2,4) => cd

-
1. `String stringValueOf = String.valueOf('c'); // most efficient`
 2. `String stringValueOfCharArray = String.valueOf(new char[]{x});`
 3. `String characterToString = Character.toString('c');`
 4. `String characterObjectToString = new Character('c').toString();`
 5. `String concatBlankString = 'c' + "";`
 6. `String fromCharArray = new String(new char[]{x});`

문제 12) Longest Substring With At Most Two Distinct

Problem

Given a string s , find the length of the longest substring t that contains at most 2 distinct characters.

Example 1:

Input: "eceba"

Output: 3

Explanation: t is "ece" which its length is 3.

Example 2:

Input: "ccaabbb"

Output: 5

Explanation: t is "aabbb" which its length is 5.

Longest Substring With At Most Two Distinct

Problem

Map, two pointer, Math.max

Input : String s = "ccaabbb"

Output: 5

"aabbb" which its length is 5.

solution

1. End-start 로 비교한 값을 length에 담는다.
2. 문자를 2개 인식하기 위해서 Counter로 담는다
이때 counter는 Map

0 1 2 3 4 5 6

c c a a b b b

length start – end

Counter: 문자 갯수

문제 13) Maximum Subarray

Problem

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

Example:

Input: `[-2,1,-3,4,-1,2,1,-5,4]`,

Output: 6

Explanation: `[4,-1,2,1]` has the largest sum = 6.

Maximum Subarray

Problem

Subarray 중에 합이 제일 큰 수, 1 두개의 합 2 nums[i]

Example

0 1 2 3 4 5 6 7 8

int[] nums = { -2, 1, -3, 4, -1, 2, 1, -5, 4 }

Output: 6 Math.max(newSum+nums[i], nums[i]), Math.max(max, newSum);

-2

-2+1= -1 newSum+nums[1]: -1 nums[1]: 1 max:1

1+-3= -2 newSum+nums[2]: -2 nums[2]: -3, max =1

-2+4= 2 newSum+nums[3]: 2 nums[3]: 4 max =4

4+-1= 3 newSum+nums[4]: 3 nums[4]: -1, max =4

3+-1= 3 newSum+nums[5]: 5 nums[5]: 2, max =5

5+-1= 3 newSum+nums[6]: 6 nums[6]: 1, max =6

6+-1= 3 newSum+nums[7]: 1 nums[7]: -5, max =6

1+4= 5 newSum+nums[8]: 5 nums[8]: 4, max =6

문제 14) Find Anagrams Mapping

Problem

Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A.

We want to find an index mapping P, from A to B. A mapping $P[i] = j$ means the i th element in A appears in B at index j .

These lists A and B may contain duplicates. If there are multiple answers, output any of them.

For example, given

A = [12, 28, 46, 32, 50]

B = [50, 12, 32, 46, 28]

We should return

[1, 4, 3, 2, 0]

as $P[0] = 1$ because the 0th element of A appears at B[1], and $P[1] = 4$ because the 1st element of A appears at B[4], and so on.

Note:

A, B have equal lengths in range [1, 100].

$A[i], B[i]$ are integers in range $[0, 10^5]$.

Find Anagrams Mapping

Problem

`int[] A = {11, 27, 45, 31, 50};`

`int[] B = {50, 11, 31, 45, 27};`

Output : [1 4 3 2 0]

Example

1. Map을 이용한다. Key, Value를 이용
2. Array B의 순서를 리턴
3. `int[]`
4. Map A의 값을 셋팅합니다.
5. 꺼낼 때는 B key 값으로 리턴된 걸 `int[]`

문제 15) Find All Anagrams

Problem

Given a string *s* and a non-empty string *p*, find all the start indices of *p*'s anagrams in *s*.

Strings consists of lowercase English letters only and the length of both strings *s* and *p* will not be larger than 20,100.

The order of output does not matter.

Example 1:

Input:

s: "cbaebabacd" *p*: "abc"

Output:

[0, 6]

Explanation:

The substring with start index = 0 is "cba", which is an anagram of "abc".

The substring with start index = 6 is "bac", which is an anagram of "abc".

Find All Anagrams

Problem

String txt = "BACDGABCD A";

String pat = "ABCD";

Output : [0, 5, 6]

Example

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

B A C D G A B C D A



1. Pattern : ABCD의 아스키값을 Array에 담는다. 65,66,67,68
2. 대상소스(txt)를 비교한다. 이중 for문으로
3. Txt, pat을 같은지 비교한다.

문제 16) SpiralMatrix

Problem

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

Example 1:

Input:

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

Output: [1,2,3,6,9,8,7,4,5]

SpiralMatrix

Problem

{ 1, 2, 3 },
{ 4, 5, 6 },
{ 7, 8, 9 }

Output {1,2,3,6,9,8,7,4,5}

→
{ 00, 01, 02, 03}, 규칙 찾기

{ 10, 11, 12, 13},

{ 20, 21, 22, 23 }

Solution

```
int rowStart = 0;  
int rowEnd   = 2 matrix.length-1;  
  
int colStart = 0;  
int colEnd   = 3 matrix[0].length - 1;
```

→ ←

{ 00, 01, 02, 03}, { 20, 21, 22, 23},
[rowStart][colEnd] [rowEnd][colEnd]
rowStart++ rowEnd--

03 ↓ ↑ 10
13 20
23

[rowStart][colEnd] [rowEnd][colStart]
colEnd-- colStart++

문제 17) Group Anagrams

Problem

Given an array of strings, group anagrams together.

Example:

Input: ["eat", "tea", "tan", "ate", "nat", "bat"],

Output:

```
[  
  ["ate","eat","tea"],  
  ["nat","tan"],  
  ["bat"]  
]
```

Note:

All inputs will be in lowercase.

The order of your output does not matter.

Group Anagrams

Problem

Input: ["eat", "tea", "tan", "ate", "nat", "bat"],

Output:

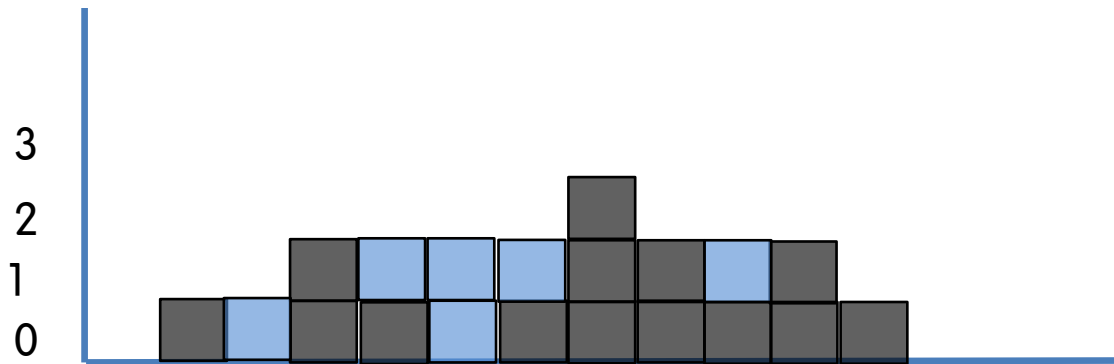
```
[  
  ["ate","eat","tea"],  
  ["nat","tan"],  
  ["bat"]  
]
```

Example

1. Map 이용
2. Anagrams -> 소팅 (key , List<String>)

문제 18) Trapping Rain Water

Problem



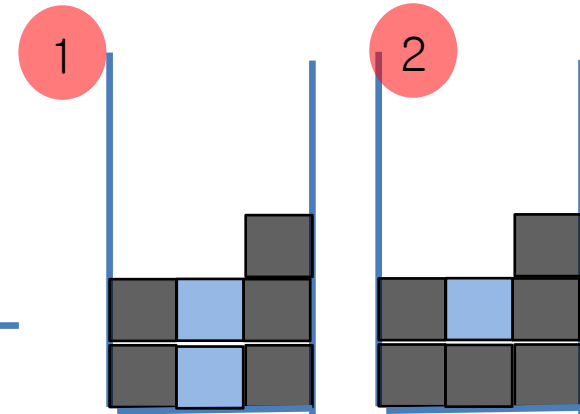
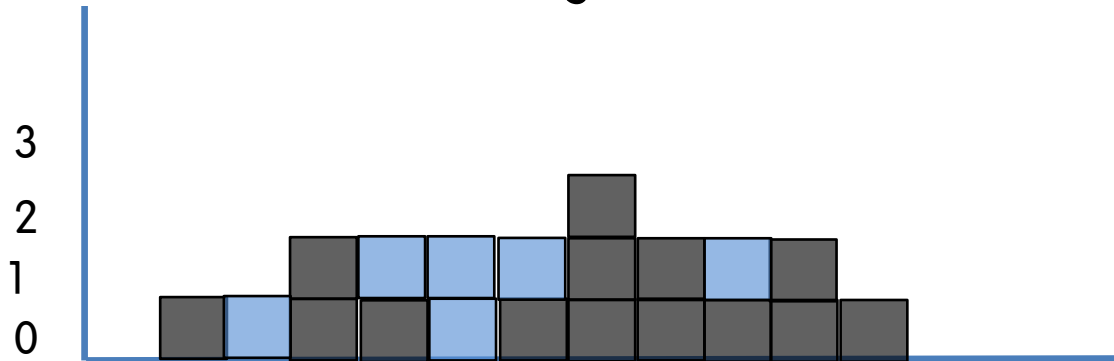
Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

The above elevation map is represented by array `[0,1,0,2,1,0,1,3,2,1,2,1]`. In this case, 6 units of rain water (blue section) are being trapped. **Thanks Marcos** for contributing this image!

Trapping Rain Water

Problem

Height : {0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1}



Example

1. int[] left: {0, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3}
2. int[] right: {3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 1}
3. Math.min : {0, 1, 1, 2, 2, 2, 2, 3, 2, 2, 2, 1}
4. Height : {0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1}
5. 0, 0, 1, 0, 1, 2, 1, 0, 0, 1, 0, 0
1. Math.min(left[i], right[i]) - height[i];

설명

1) 물을 부었을 때 높이 2
왼쪽벽 2, 오른쪽벽 3
작은값 2

설명

2) 물을 부었을 때 높이 1
왼쪽벽 2, 오른쪽벽 3
작은값 2 - 자체높이 1 = 1

문제 19) Kth Largest Element In An Array

Problem

Find the kth largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Example 1:

Input: [3,2,1,5,6,4] and $k = 2$

Output: 5

Example 2:

Input: [3,2,3,1,2,4,5,5,6] and $k = 4$

Output: 4

Note:

You may assume k is always valid, $1 \leq k \leq \text{array's length}$.

Kth Largest Element In An Array

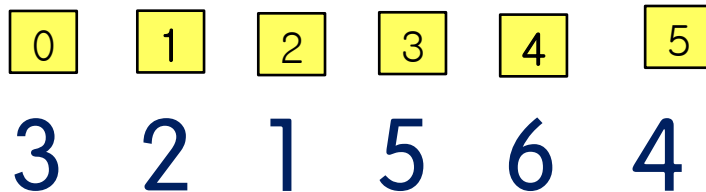
Problem

```
int[] nums = {3,2,1,5,6,4};  
int k =2;
```

Output: 5

solution

1. 배열 오름차순 {1,2,3,4,5,6} $6-2=4$
2. PriorityQueue , k=2



5

6

문제 20) Missing Ranges

Problem

Given a sorted integer array `nums`, where the range of elements are in the inclusive range `[lower, upper]`, return its missing ranges.

Example:

Input: `nums = [0, 1, 3, 50, 75]`, `lower = 0` and `upper = 99`,
Output: `["2", "4->49", "51->74", "76->99"]`

Missing Ranges

Problem

```
int[] nums = {2, 3, 5, 50, 75};  
int lower = 0 ,upper = 99;
```

Output: [0->1, 4, 6->49, 51->74, 76->99]

solution

1. $0 < 2$: $\text{lower} < \text{nums}[0]$, 0->1
2. $2 < 3$, $3 < 5$ $\text{nums}[i] + 1 < \text{nums}[i+1]$
2~3 인 경우 Pass
3~5 인 경우 4 , $3+1=5-1$ $\text{nums}[i]+1, \text{nums}[i+1]-1$
3. $75 < 99$ $\text{nums}[\text{nums.length}-1] < \text{upper}$
76<99 $\text{nums}[\text{nums.length}-1]+1, \text{upper}$