

리트코드 세미나 1차시

2023년도 1학기 한양대 전전컴 학술통아리 바라미의 금융세미나
2019024439 신현

Be Artists of Robotics &
Advanced Micro Intelligence

바라미

Since 1994

목차

1차시의 목표! 리트코드가 무엇인지 알아보자~~

- 작년 세미나와 올해 세미나의 차이점
- 리트코드 Easy 문제를 제출해보며 리트코드와 친해지기
 - Easy 문제 소개 / 풀이 / 코드 / 제출하는 방법 리트코드 위클리 소개
- 리트코드 Biweekly 및 여러 Easy – Medium 문제 풀이

작년 세미나와 올해 세미나의 차이점

1. BOJ 가 아니라 리트코드임

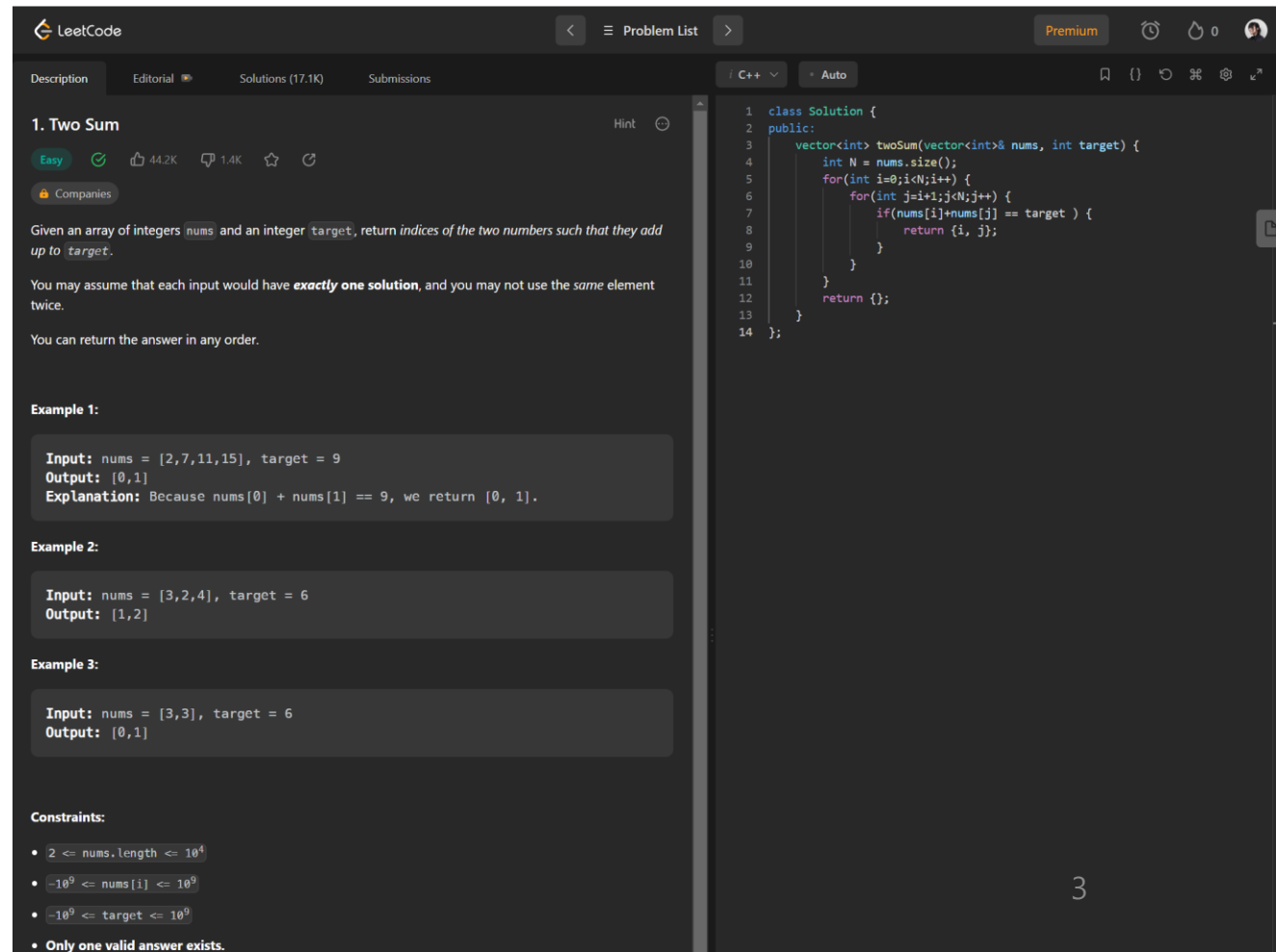
- 리트코드는 웹으로 접속해서 코드를 작성하고 컴파일 할 수 있어서
에디터에 대해 설명하느라 시간이 가지도 않고,
여러분도 쉽게 접근 할 수 있다.

2. 발표자료가 LaTeX 이 아니라 PPT임

- 예쁘진 않아도 편하더라구요...
- 단점: 코드 Ctrl-C, V가 잘 안됨
- 코드는 깃헙 code 파일에 올려드림...

3. 변수 선언, for문, if 문 이런 문법 설명 안할 예정

- c++세미나가 바라미가 오래되었기 때문에
전통이라 매년 하던 거긴 한데 ,
솔직히 요즘 애들 C++던 python이든 배우고 오거나
굳이 배우지 않았어도 학교에서 1~2주면 다 배움...



리트코드에 가입하기

1. 아래 사이트에 들어가서 가입을 해주세요!
<https://leetcode.com/>

리트코드 첫 번째 문제를 풀어보자

- <https://leetcode.com/problems/two-sum/> 에 접속해주세요.
- 문제: 숫자 배열(nums)과 target이 주어졌을 때, 숫자 배열에서 두 인덱스를 골라, 해당 인덱스의 두 숫자의 합이 타겟이 되는 인덱스의 위치를 반환한다.
- 입력으로는 답은 하나 이상이 있음이 보장 되는 입력만 들어온다.

온라인 저지는 코드를 제출하면 정해진 입력을 넣었을 때, 여러분의 코드가 정해진 출력을 내는지를 확인하여 코드가 맞았는지 판별합니다.

입출력 예시

입력(Input): nums = [2, 7, 11, 15] / target = 9 라고 주어지면,

출력(Output): 답은 0, 1 입니다.

(배열의 0번 인덱스의 2와 배열의 1번 인덱스의 숫자인 7의 합이 9입니다.)

풀이

- 첫 번째 문제에 대해선 총 세 가지 언어로 풀이를 작성하였습니다.
- 꽤 유명한 언어들인 C/C++/Python 이렇게 세 가지로 작성하였고,
- 첫 번째 문제라 세 가지버전을 모두 첨부하였지만
- 앞으로의 문제에는 C++/Python만 첨부하거나 C++만 첨부할 수도 있습니다... (C언어 포인터 귀찮음...)

첫 번째 문제의 답 (C)

```
int* twoSum(int* nums, int numsSize, int target,
int* returnSize){
    *returnSize = 2;
    int* answer = (int*)malloc(2*sizeof(int));
    for(int i=0;i<numsSize;i++) {
        for(int j=i+1;j<numsSize;j++) {
            if(nums[i]+nums[j] == target ) {
                answer[0] = i;
                answer[1] = j;
                return answer;
            }
        }
    }
    // 실행되진 않지만 컴파일 때문에 추가한 코드
    answer[0] = -1;
    answer[1] = -1;
    return answer;
}
```

첫 문제라 C언어 코드도 추가하긴 했는데
저는 C언어를 별로 좋아하지 않고
C++를 많이 씁니다.

혹여 C언어만 알고 C++를 아직 모르지만
배워보고 싶은 분이 있다면

이 다음 페이지의 C++ 코드를 보고
vector<int> 라는 짱짱 편한 stl을 공부하시면

문제 풀 때 편합니다.

C++는 대충 C언어 코드도 어지간한건 다 지원되지만
C언어에 편리한 함수들을 추가해놓은 확장판(?) 정도로
생각하심 편합니다.

첫 번째 문제의 답 (C++)

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        int N = nums.size();
        for(int i=0; i<N; i++) {
            for(int j=i+1; j<N; j++) {
                if(nums[i]+nums[j] == target ) {
                    return {i, j};
                }
            }
        }
        return {}; // 입력에서 답이 있는게 보증된다 하더라도 컴파일 되려면 {} 리턴해야함.
    }
};
```


첫 번째 문제의 답 (Python)

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        N = len(nums)
        for i in range(0, N):
            for j in range(i+1, N):
                if nums[i] + nums[j] == target:
                    return [i, j]
```

여기서 `nums: List[int]` 이런 식으로 변수 옆에 : 붙이고 `List[int]` 처럼 생긴 것은 type hinting 이라고 해서 Python 3.9 버전부터 지원되는 것인데, 컴파일 결과에 영향을 주는 것이 아니라서 코드 짤 때는 크게 신경쓰지 않으셔도 됩니다!

하지만 이 내용을 보면 입력의 자료형이 무엇인지, 출력으로는 어느 자료형을 리턴해야 하는지 알 수 있습니다!

제출해보기

LeetCode

Problem List

Premium

1. Two Sum

Easy

44.2K

1.4K

Companies

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

Input: `nums = [2,7,11,15]`, `target = 9`
Output: `[0,1]`
Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: `nums = [3,2,4]`, `target = 6`
Output: `[1,2]`

Example 3:

Input: `nums = [3,3]`, `target = 6`
Output: `[0,1]`

Constraints:

```
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4         int N = nums.size();
5         for(int i=0; i<N; i++) {
6             for(int j=i+1; j<N; j++) {
7                 if(nums[i]+nums[j] == target) {
8                     return {i, j};
9                 }
10            }
11        }
12        return {};
13    }
14};
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

nums =
[2,7,11,15]

target =

Console Run Submit

1. Run 을 누르면 예제를 돌려 볼 수 있습니다.

2. Submit 을 누르면 전체 테스트케이스에 대해서 채점을 해볼 수 있습니다.

Run과 Submit의 차이점

- Run 은 예제(일부 테스트케이스)에 대해 돌려보는 것을 의미한다.
- Submit 은 전체 테스트케이스에 대해 돌려보는 것을 의미한다.

그냥 문제 풀 때는 별로 상관 없긴 한데,

- 위클리(시간 제한 있는 연습)에서는 랭킹을 매길 때,
문제를 푸는데 걸린 시간 + 오답인 Submit 한 횟수 * 5분으로 랭킹을 매기기 때문에

Run을 통해 일부 테스트케이스가 다 통과하는지 확인하고
Submit은 신중하게 하는 것이 좋다.

문제를 아무리 생각해도 답을 잘 모르겠다면

- 문제에서 Solutions 를 누르면 다른 사람들이 올린 게시글들과 해당 게시글들의 코드를 볼 수 있다! (그렇다고 그것을 베끼는 것은 추천하지 않습니다.)

The screenshot shows the LeetCode interface for problem 2578. The 'Solutions (303)' tab is selected and highlighted with a red box. Below the tabs, there is a search bar and a '+ Solution' button. A list of solutions is displayed, with the first three highlighted by a red box:

- 2578: Solution with step by step explanation** by Marlen09 (Mar 05, 2023) in Python/Python3. 5 upvotes, 565 views, 0 comments.
- [JavaScript] 2578. Split With Minimum Sum** by pgmreddy (Mar 05, 2023) in JavaScript. 9 upvotes, 176 views, 1 comment.
- Explained - Sorting of digit || Very Simple and Easy to Understand Solution** by kreakEmp (Mar 05, 2023) in C++. 22 upvotes, 2K views, 2 comments.

The right side of the screen shows a C++ code editor with the following code:

```
1 class Solution {
2 public:
3     int splitNum(int num) {
4         vector<int> nums;
5         while(num > 0) {
6             nums.push_back(num % 10);
7             num /= 10;
8         }
9         int one = 0, two = 0;
10        sort(nums.begin(), nums.end());
11        for(int i = 0; i < nums.size(); i++) {
12            if(i%2 == 0) one = one * 10 + nums[i];
13            else two = two * 10 + nums[i];
14        }
15        return one + two;
16    }
17};
```

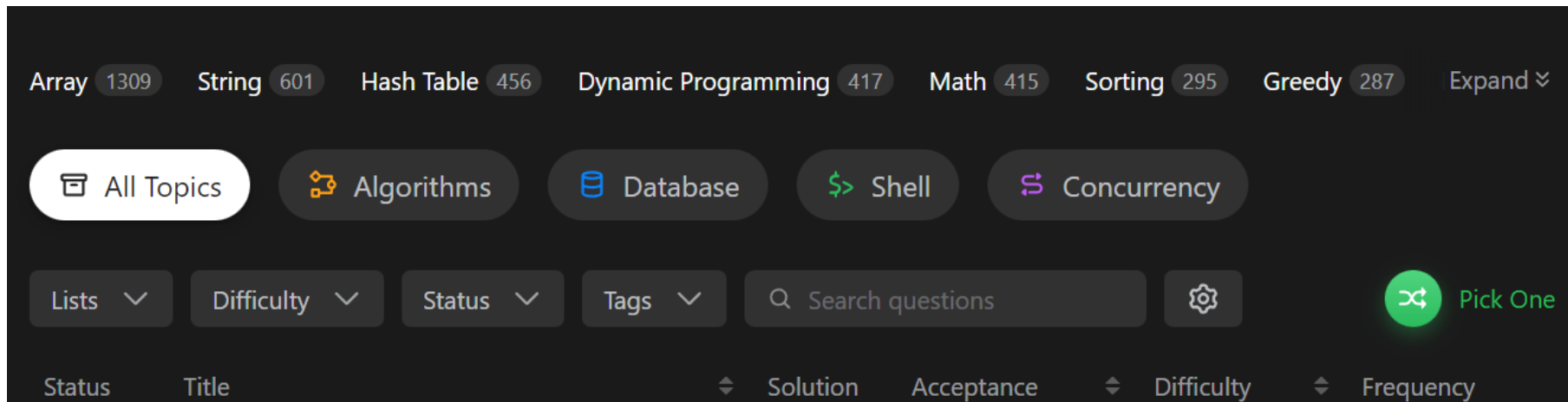
리트코드에선 어떤 문제를 풀어야 하는가

1. 여러분이 아직 뉴비라면

이렇게 Easy 문제만 골라서 볼 수도 있습니다!

<https://leetcode.com/problemset/all/?difficulty=EASY&page=1>

그 외에도 알고리즘 종류 등으로도 필터를 걸 수 있으니 원하는 종류의 문제를 잘 찾아서 풀어보세요!



리트코드에선 어떤 문제를 풀어야 하는가

2. Medium 문제도 썸 풀만하다면, 위클리 참가!

격주 토요일 오후 11시 30분 ~ 오후 1시
매주 일요일 오전 11시 30분 ~ 오후 1시

저도 (다른 온라인 저지로는 문제를 많이 풀었다고는 해도) 리트코드는 최근에 시작하였기 때문에 사실 오늘 소개한 Medium, Hard 문제는, 최근 2주 전의 위클리 문제였습니다 ㅎㅎㅎ...
(문제 쫌 좋은거 가져오진 못한거같기도...)

위클리에 참여하면, 아마 그 다음 세미나 시간에서 그 문제 풀이해줄지도!?

그 외 디코에서 실시간으로 설명한 문제
들

<https://leetcode.com/contest/biweekly-contest-99/problems/split-with-minimum-sum/>

EASY

```
class Solution {
public:
    int splitNum(int num) {
        vector<int> nums;
        while(num > 0) {
            nums.push_back(num % 10);
            num /= 10;
        }
        reverse(nums.begin(), nums.end())
        int one = 0, two = 0;
        sort(nums.begin(), nums.end());
        for(int i = 0; i < nums.size(); i++) {
            if(i%2 == 0) one = one * 10 + nums[i];
            else two = two * 10 + nums[i];
        }
        return one + two;
    }
};
```


<https://leetcode.com/contest/biweekly-contest-99/problems/count-total-number-of-colored-cells/>

- Medium

```
class Solution:
    def coloredCells(self, n: int) -> int:
        ans = 1
        for i in range(0, n):
            ans = ans + (i * 4)
        return ans
```

<https://leetcode.com/contest/biweekly-contest-100/problems/find-score-of-an-array-after-marking-all-elements/>

Medium

```
#define ll long long
#define pii pair<int, int>
class Solution {
public:
    long long findScore(vector<int>& nums) {
        int N = nums.size();
        vector<pii> arr;
        vector<bool> used;
        used.assign(N, false);
        ll ans = 0;
        for(int i=0;i<N;i++) {
            arr.push_back({nums[i], i});
        }
        sort(arr.begin(), arr.end());
        for(pii element:arr) {
            if(used[element.second]) continue;
            else {
                used[element.second] = true;
                if(element.second - 1 >= 0)
                    used[element.second-1] = true;
                if(element.second + 1 < N)
                    used[element.second + 1] = true;
                ans += (ll)element.first;
            }
        }
        return ans;
    }
};
```

<https://leetcode.com/problems/palindrome-number/>

EASY

```
class Solution:
    def isPalindrome(self, x: int) -> bool:
        if x < 0:
            return False

        x_str = list(map(int, str(x)))

        N = len(x_str)

        left = 0
        right = N - 1
        while left <= right:
            if x_str[left] != x_str[right]:
                return False
            left += 1
            right -= 1

        return True
```

<https://leetcode.com/problems/plus-one/>

EASY

```
class Solution:
    def plusOne(self, digits: List[int]) ->
List[int]:
    digits.reverse()
    ans = []

    is_nine = True
    for digit in digits:
        if is_nine and digit == 9:
            ans.append(0)
        else:
            if is_nine:
                digit += 1
            is_nine = False
            ans.append(digit)

    if is_nine:
        ans.append(1)

    ans.reverse()
    return ans
```