

취업 준비 스터디 7주차 멘토링

2022.11.09



발표자 : 박동학

1. 소통해요!
2. 이번주 문제 해설
3. 개발 지식
4. 운영체제
5. 네트워크
6. QnA

진행 상황 공유 [중간 점검]

- 업계?
- 직무?
- 서류?
- 코딩테스트?
- 면접?



KB 국민카드

Why 국민카드 ?

- 업계?
- 직무?
- 전형 절차는 ?
- 다른 기업들은 어떤가?

1. 파도반 수열

파도반 수열

성공

다국어

☆

한국어 ▾

3 실버 III

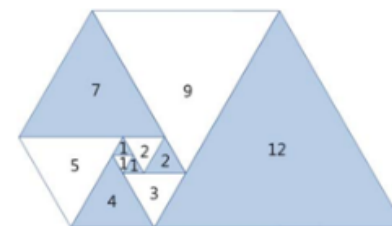
시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	74857	32803	26885	42.490%

문제

오른쪽 그림과 같이 삼각형이 나선 모양으로 놓여져 있다. 첫 삼각형은 정삼각형으로 변의 길이는 1이다. 그 다음에는 다음과 같은 과정으로 정삼각형을 계속 추가한다. 나선에서 가장 긴 변의 길이를 k 라 했을 때, 그 변에 길이가 k 인 정삼각형을 추가한다.

파도반 수열 $P(N)$ 은 나선에 있는 정삼각형의 변의 길이이다. $P(1)$ 부터 $P(10)$ 까지 첫 10개 숫자는 1, 1, 1, 2, 2, 3, 4, 5, 7, 9이다.

N 이 주어졌을 때, $P(N)$ 을 구하는 프로그램을 작성하시오.



입력

첫째 줄에 테스트 케이스의 개수 T 가 주어진다. 각 테스트 케이스는 한 줄로 이루어져 있고, N 이 주어진다. ($1 \leq N \leq 100$)

출력

각 테스트 케이스마다 $P(N)$ 을 출력한다.

예제 입력 1 복사

```
2
6
12
```

예제 출력 1 복사

```
3
16
```

규칙 찾기 문제

2. 적록색약

문제

적록색약은 빨간색과 초록색의 차이를 거의 느끼지 못한다. 따라서, 적록색약인 사람이 보는 그림은 아닌 사람이 보는 그림과는 좀 다를 수 있다.

크기가 $N \times N$ 인 그리드의 각 칸에 R(빨강), G(초록), B(파랑) 중 하나를 색칠한 그림이 있다. 그림은 몇 개의 구역으로 나뉘어져 있는데, 구역은 같은 색으로 이루어져 있다. 또, 같은 색상이 상하좌우로 인접해 있는 경우에 두 글자는 같은 구역에 속한다. (색상의 차이를 거의 느끼지 못하는 경우도 같은 색상이라 한다)

예를 들어, 그림이 아래와 같은 경우에

```
RRRBB
GGBBB
BBBRR
BBRRR
RRRRR
```

적록색약이 아닌 사람이 봤을 때 구역의 수는 총 4개이다. (빨강 2, 파랑 1, 초록 1) 하지만, 적록색약인 사람은 구역을 3개 볼 수 있다. (빨강-초록 2, 파랑 1)

그림이 입력으로 주어졌을 때, 적록색약인 사람이 봤을 때와 아닌 사람이 봤을 때 구역의 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N 이 주어진다. ($1 \leq N \leq 100$)

둘째 줄부터 N 개 줄에는 그림이 주어진다.

출력

적록색약이 아닌 사람이 봤을 때의 구역의 개수와 적록색약인 사람이 봤을 때의 구역의 수를 공백으로 구분하여 출력한다.

예제 입력 1 복사

```
5
RRRBB
GGBBB
BBBRR
BBRRR
RRRRR
```

예제 출력 1 복사

```
4 3
```

구역 나누기 - BFS문제

3. 동전 0

동전 0 성공



4 실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	99751	52247	40491	51.821%

문제

준규가 가지고 있는 동전은 총 N 종류이고, 각각의 동전을 매우 많이 가지고 있다.

동전을 적절히 사용해서 그 가치의 합을 K 로 만들려고 한다. 이때 필요한 동전 개수의 최솟값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N 과 K 가 주어진다. ($1 \leq N \leq 10$, $1 \leq K \leq 100,000,000$)

둘째 줄부터 N 개의 줄에 동전의 가치 A_i 가 오름차순으로 주어진다. ($1 \leq A_i \leq 1,000,000$, $A_1 = 1$, $i \geq 2$ 인 경우에 A_i 는 A_{i-1} 의 배수)

출력

첫째 줄에 K 원을 만드는데 필요한 동전 개수의 최솟값을 출력한다.

예제 입력 1 [복사](#)

```
10 4200
1
5
10
50
100
500
1000
5000
10000
50000
```

예제 출력 1 [복사](#)

```
6
```

전형적인 그리디!

왜 성립?

4. 최대 힙

최대 힙

성공



2 실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초 (추가 시간 없음) (하단 참고)	256 MB	53028	24025	18791	46.676%

문제

널리 잘 알려진 자료구조 중 최대 힙이 있다. 최대 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성하시오.

- 배열에 자연수 x 를 넣는다.
- 배열에서 가장 큰 값을 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N ($1 \leq N \leq 100,000$)이 주어진다. 다음 N 개의 줄에는 연산에 대한 정보를 나타내는 정수 x 가 주어진다. 만약 x 가 자연수라면 배열에 x 라는 값을 넣는(추가하는) 연산이고, x 가 0이라면 배열에서 가장 큰 값을 출력하고 그 값을 배열에서 제거하는 경우이다. 입력되는 자연수는 2^{31} 보다 작다.

출력

입력에서 0이 주어진 회수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 큰 값을 출력하라고 한 경우에는 0을 출력하면 된다.

Just Heapq

절댓값 힙

성공

1 실버 I

시간 제한	메모리 제한	제출	정답
1 초 (추가 시간 없음) (하단 참고)	256 MB	30243	16916

```
>>> Q = []
>>> heapq.heappush(Q, [5, -1])
>>> heapq.heappush(Q, [5, 1])
>>> heapq.heappush(Q, [234234, -234234])
>>> print(heapq.heappop(Q))
[5, -1]
>>> print(heapq.heappop(Q))
[5, 1]
>>> print(heapq.heappop(Q))
[234234, -234234]
```

문제

절댓값 힙은 다음과 같은 연산을 지원하는 자료구조이다.

- 배열에 정수 x ($x \neq 0$)를 넣는다.
- 배열에서 절댓값이 가장 작은 값을 출력하고, 그 값을 배열에서 제거한다. 절댓값이 가장 작은 값이 여러개일 때는, 가장 작은 수를 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N ($1 \leq N \leq 100,000$)이 주어진다. 다음 N 개의 줄에는 연산에 대한 정보를 나타내는 정수 x 가 주어진다. 만약 x 가 0이 아니라면 배열에 x 라는 값을 넣는(추가하는) 연산이고, x 가 0이라면 배열에서 절댓값이 가장 작은 값을 출력하고 그 값을 배열에서 제거하는 경우이다. 입력되는 정수는 -2^{31} 보다 크고, 2^{31} 보다 작다.

출력

입력에서 0이 주어진 회수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 절댓값이 가장 작은 값을 출력하라고 한 경우에는 0을 출력하면 된다.

Just Heapq

Python Heapq의 정렬기준은 ?

6. ATM

ATM

성공



4 실버 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	76969	51506	41713	67.484%

문제

인하은행에는 ATM이 1대밖에 없다. 지금 이 ATM앞에 N 명의 사람들이 줄을 서있다. 사람은 1번부터 N 번까지 번호가 매겨져 있으며, i 번 사람이 돈을 인출하는데 걸리는 시간은 P_i 분이다.

사람들이 줄을 서는 순서에 따라서, 돈을 인출하는데 필요한 시간의 합이 달라지게 된다. 예를 들어, 총 5명이 있고, $P_1 = 3$, $P_2 = 1$, $P_3 = 4$, $P_4 = 3$, $P_5 = 2$ 인 경우를 생각해보자. [1, 2, 3, 4, 5] 순서로 줄을 선다면, 1번 사람은 3분만에 돈을 뽑을 수 있다. 2번 사람은 1번 사람이 돈을 뽑을 때 까지 기다려야 하기 때문에, $3+1 = 4$ 분이 걸리게 된다. 3번 사람은 1번, 2번 사람이 돈을 뽑을 때까지 기다려야 하기 때문에, 총 $3+1+4 = 8$ 분이 필요하게 된다. 4번 사람은 $3+1+4+3 = 11$ 분, 5번 사람은 $3+1+4+3+2 = 13$ 분이 걸리게 된다. 이 경우에 각 사람이 돈을 인출하는데 필요한 시간의 합은 $3+4+8+11+13 = 39$ 분이 된다.

줄을 [2, 5, 1, 4, 3] 순서로 줄을 서면, 2번 사람은 1분만에, 5번 사람은 $1+2 = 3$ 분, 1번 사람은 $1+2+3 = 6$ 분, 4번 사람은 $1+2+3+3 = 9$ 분, 3번 사람은 $1+2+3+3+4 = 13$ 분이 걸리게 된다. 각 사람이 돈을 인출하는데 필요한 시간의 합은 $1+3+6+9+13 = 32$ 분이다. 이 방법보다 더 필요한 시간의 합을 최소로 만들 수는 없다.

줄을 서 있는 사람의 수 N 과 각 사람이 돈을 인출하는데 걸리는 시간 P_i 가 주어졌을 때, 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 사람의 수 N ($1 \leq N \leq 1,000$)이 주어진다. 둘째 줄에는 각 사람이 돈을 인출하는데 걸리는 시간 P_i 가 주어진다. ($1 \leq P_i \leq 1,000$)

출력

첫째 줄에 각 사람이 돈을 인출하는데 필요한 시간의 합의 최솟값을 출력한다.

예제 입력 1 복사

```
5
3 1 4 3 2
```

그리디 & 스케줄링

예제 출력 1 복사

```
32
```

7. 경로 찾기

경로 찾기

성공



1 실버 I

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	36515	21489	15722	58.509%

문제

가중치 없는 방향 그래프 G 가 주어졌을 때, 모든 정점 (i, j) 에 대해서, i 에서 j 로 가는 경로가 있는지 없는지 구하는 프로그램을 작성하시오.

입력

첫째 줄에 정점의 개수 N ($1 \leq N \leq 100$)이 주어진다. 둘째 줄부터 N 개 줄에는 그래프의 인접 행렬이 주어진다. i 번째 줄의 j 번째 숫자가 1인 경우에는 i 에서 j 로 가는 간선이 존재한다는 뜻이고, 0인 경우는 없다는 뜻이다. i 번째 줄의 i 번째 숫자는 항상 0이다.

출력

총 N 개의 줄에 걸쳐서 문제의 정답을 인접행렬 형식으로 출력한다. 정점 i 에서 j 로 가는 경로가 있으면 i 번째 줄의 j 번째 숫자를 1로, 없으면 0으로 출력해야 한다.

예제 입력 1 복사

```
3
0 1 0
0 0 1
1 0 0
```

예제 출력 1 복사

```
1 1 1
1 1 1
1 1 1
```

단순 경로 찾기

LCS 2

스페셜 저지



4 골드 IV

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
0.1 초 (하단 참고)	256 MB	27248	9667	7454	38.200%

문제

LCS(Longest Common Subsequence, 최장 공통 부분 수열)문제는 두 수열이 주어졌을 때, 모두의 부분 수열이 되는 수열 중 가장 긴 것을 찾는 문제이다.

예를 들어, ACAYKP와 CAPCAK의 LCS는 ACAK가 된다.

입력

첫째 줄과 둘째 줄에 두 문자열이 주어진다. 문자열은 알파벳 대문자로만 이루어져 있으며, 최대 1000글자로 이루어져 있다.

출력

첫째 줄에 입력으로 주어진 두 문자열의 LCS의 길이를, 둘째 줄에 LCS를 출력한다.

LCS가 여러 가지인 경우에는 아무거나 출력하고, LCS의 길이가 0인 경우에는 둘째 줄을 출력하지 않는다.

예제 입력 1 [복사](#)

```
ACAYKP
CAPCAK
```

예제 출력 1 [복사](#)

```
4
ACAK
```

LCS 문제

(Known Algorithm)

이진탐색

이진 탐색이란 ? → 검색 범위를 줄여 나가면서 원하는 데이터를 찾는 알고리즘

탐색 ? → 순차 탐색 ? → 이분/이진 탐색 ?

예시를 들어봅시다.

크기가 100만인 배열이 있다.

안에는 서로 다른 정수가 무작위로 들어가 있고 나는 1,000,001를 찾는다.

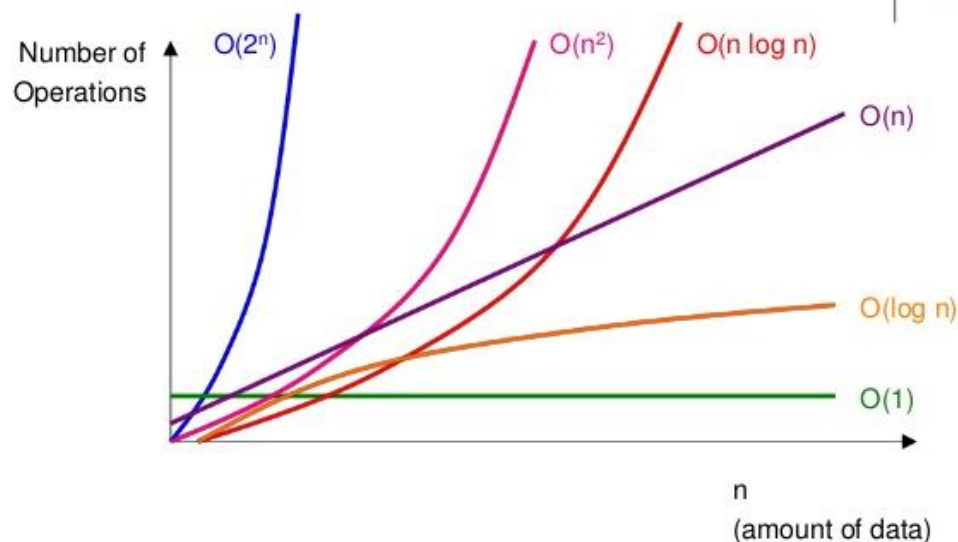
하나하나씩 검사하면 수행 시간은 ? → $O(N)$ → 1000만번의 연산

100억 개면 → 100억번의 연산

C언어 기준으로 연산 10억번이면 1초이상 소요

Python의 경우 훨씬 느림

Comparing Big O Functions



좋음(Better)

나쁨(Worse)

순위	명칭
$O(1)$	상수 시간(Constant time)
$O(\log N)$	로그 시간(Log time)
$O(N)$	선형 시간
$O(N \log N)$	로그 선형 시간
$O(N^2)$	이차 시간
$O(N^3)$	삼차 시간
$O(2^n)$	지수 시간

Revisit 랜선 자르기 문제

랜선 자르기

성공



2 실버 II

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	140930	32732	22083	21.110%

문제

집에서 시간을 보내던 오영식은 박성원의 부름을 받고 급히 달려왔다. 박성원이 캠프 때 쓸 N 개의 랜선을 만들어야 하는데 너무 바빠서 영식에게 도움을 청했다.

이미 오영식은 자체적으로 K 개의 랜선을 가지고 있다. 그러나 K 개의 랜선은 길이가 제각각이다. 박성원은 랜선을 모두 N 개의 같은 길이의 랜선으로 만들고 싶었기 때문에 K 개의 랜선을 잘라서 만들어야 한다. 예를 들어 300cm 짜리 랜선에서 140cm 짜리 랜선을 두 개 잘라내면 20cm는 버려야 한다. (이미 자른 랜선은 붙일 수 없다.)

편의를 위해 랜선을 자르거나 만들 때 손실되는 길이는 없다고 가정하며, 기존의 K 개의 랜선으로 N 개의 랜선을 만들 수 없는 경우는 없다고 가정하자. 그리고 자를 때는 항상 센티미터 단위로 정수길이만큼 자른다고 가정하자. N 개보다 많이 만드는 것도 N 개를 만드는 것에 포함된다. 이때 만들 수 있는 최대 랜선의 길이를 구하는 프로그램을 작성하시오.

입력

첫째 줄에는 오영식이 이미 가지고 있는 랜선의 개수 K , 그리고 필요한 랜선의 개수 N 이 입력된다. K 는 1이상 10,000이하의 정수이고, N 은 1이상 1,000,000이하의 정수이다. 그리고 항상 $K \leq N$ 이다. 그 후 K 줄에 걸쳐 이미 가지고 있는 각 랜선의 길이가 센티미터 단위의 정수로 입력된다. 랜선의 길이는 $2^{31}-1$ 보다 작거나 같은 자연수이다.

출력

첫째 줄에 N 개를 만들 수 있는 랜선의 최대 길이를 센티미터 단위의 정수로 출력한다.

Revisit 랜선 자르기 문제

K개의 랜선을 N개의 같은 길이의 랜선으로 만들고 싶다.

$K < 10,000$ // $N < 1,000,000$

HOW?

K개의 랜선을 길이 length으로 만들어 보고 N개 이상인지 보면 된다.

→ K개의 랜선을 길이 1, 2, 3, 4, 5, ... Length까지 검사한다.

→ 순차 탐색 시 : $K_{\max} * N_{\max} = 10,000,000,000$ (100억개) → C기준 10초이상

Revisit 랜선 자르기 문제

→ 순차 탐색 시 : $K_{\max} * N_{\max} = 10,000,000,000$ (100억개)

→ C기준 10초이상

만약 K가 100만 N이 100만이라면 ? 1조개의 연산 → 1000초 = 10분이상

Binary Search 라면 ? → Binary Search $O(\log N)$ [밑이 2인] → $O(N \log N)$

$\log(2) = 1$ / $\log(2^2) = 2$

컴퓨터에서는 2^{10} 을 1000으로 본다 // $\log(1,000,000) = 20$

따라서 $20 * 100\text{만} = 2000\text{만}$ ----→ 1초 미만

***** 10분 vs 1초

```
K, N = map(int, input().split())
lan = []
for i in range(K):
    lan.append(int(input()))
start, end = 1, max(lan)

while start <= end:
    mid = (start+end)//2
    lines = 0
    for i in lan:
        lines += i // mid

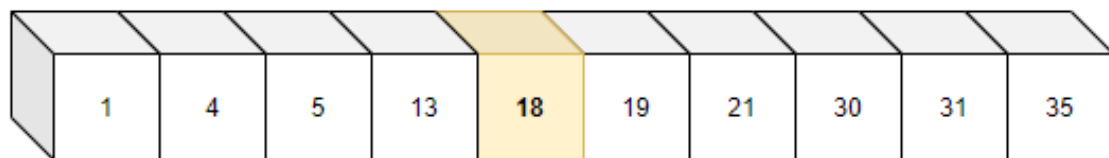
    if lines >= N:
        start = mid+1
    else:
        end = mid-1
print(end)
```

0. 오름차순으로 정렬된 배열



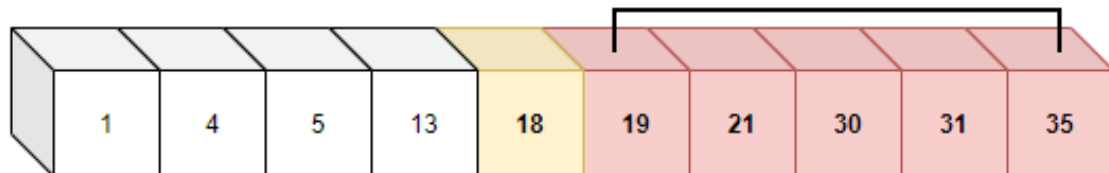
31이란 데이터를 탐색한다고 하자.
1. 배열의 **중앙값**을 선택한다. (배열의 크기 10)

low : 0
high : 9
mid : 4



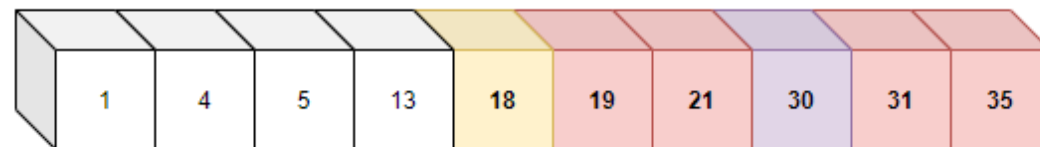
2. **18(중앙값)**과 **31**을 비교한다.
31보다 18이 작으므로 **우측요소**를 선택한다.

low : 5
high : 9



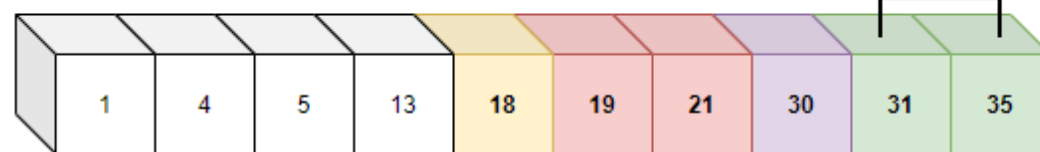
3. 선택된 우측 요소의 **중앙값**을 선택한다.

low : 5
high : 9
mid : 7



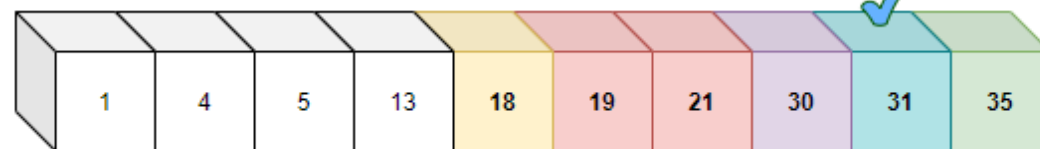
4. **30(중앙값)**과 **31**을 비교한다.
31보다 30이 작으므로 **우측요소**를 선택한다.

low : 8
high : 9



5. 남은 배열중에 **좌측요소(31)**를 선택하여 **31**과 비교한다.
31과 동일하면 탐색 종료

low : 8
high : 9
mid : 8



생활 속 이분탐색 (병뚜껑 번호 맞추기)

소주병은 1~50사이의 숫자가 적혀있다.



개발자들 끼리 술을 먹다가 병뚜껑 게임을 한다.
한명이 외친다. Binary Search!!

$\text{Log}(50) = ? 5.xxxx$

그렇다면 이 게임은 6명이상이면 필승하는 게임이다.

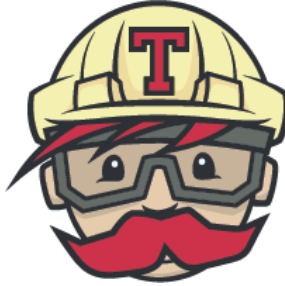
Start = 1

End = 50

What is CI/CD



Jenkins



Travis CI



circleci



GitHub Actions

What is CI/CD

CI : Continuous Integration [지속적 통합]

CI : 빌드/테스트 자동화 과정 → 보통 CI가 통과가 되지않으면 Review도 안함

→ 새로운 코드 변경에 대해서 정기적으로 이를 빌드하고 테스트 하여 끊임없이 통합할 수 있다는 개념

→ 프로젝트가 커지면 코드 변경이 많이 !! 자주 일어남 → 매번 사람이 변경점 적용 빌드, 테스트 한다면 ?

→ 주로 커밋을 기준으로 자동으로 빌드 & 테스트를 진행하고 문제가 있다면 Merge하지 않음

The screenshot displays a CI/CD dashboard with three main sections. The left section shows a summary of 'Some checks' with 3 failing and 1 successful. The middle section shows 'All checks' with 23 successful. The right section provides a detailed view of 'Some checks haven't completed yet', listing 2 failing, 1 skipped, 2 expected, and 1 successful checks. The detailed view includes a table of checks with their status, names, and details links.

Check Name	Status	Details
ARM CI / build (3.10) (pull_request)	Failing after 56m	Details
Trusted Partner PR / assign-partner-tags (pull_request_target)	Skipped	Details
AMD ROCm -- Community CI Build — rocm CI build failed	Failed	Details
Code Check - Changed Files	Expected — Waiting for status to be reported	Required
Py+CPP Test Suite - Ubuntu CPU, Python 3.9	Expected — Waiting for status to be reported	Required
cla/google	Successful in 1s — All contributors are covered under a CLA with Google	Details

What is CI/CD

CD : Continuous Deployment [지속적 배포]

CD : 배포 자동화

→ **지속적인 서비스제공**(Continuous Delivery) **or** **지속적 배포**(Continuous Deployment)를 의미

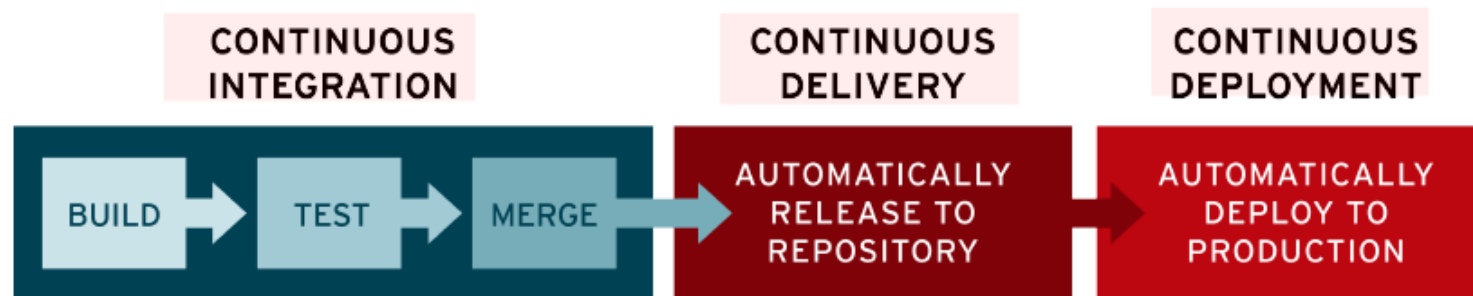
→ DevOps 방식의 극대화 -> 개발 파이프라인의 논리적 방식을 극대화

→ 코드 변경 - CI 통과 - CD를 통해 개입 없이 서비스까지 배포

→ 품질의 저하 없이 최대한 사용자에게 최신의 기능을 선사

→ Nightly Build / Nightly Deployment

→ Release를 보다 손쉽고 편안하게 한다.



RESTful API란?

{ REST API }

REST API : Representational State Transfer API

Representational State Transfer API

→ **REST**를 기반으로 만들어진 **API** (REST? API? 어지럽다..)

REST : Representational State Transfer ?

→ 자원을 이름으로 구분하여 해당 자원의 상태를 주고 받는 모든 것 (어지럽다)

- 1.HTTP URI(Uniform Resource Identifier)를 통해 자원(Resource)을 명시하고,
- 2.HTTP Method(POST, GET, PUT, DELETE, PATCH 등)를 통해
- 3.해당 자원(URI)에 대한 CRUD Operation을 적용하는 것을 의미합니다.

CRUD Operation?

CRUD → Create, Read, Update, Delete의 약자

Create : POST (데이터 생성)

Read : GET (데이터 조회)

Update : PUT, PATCH (데이터 수정)

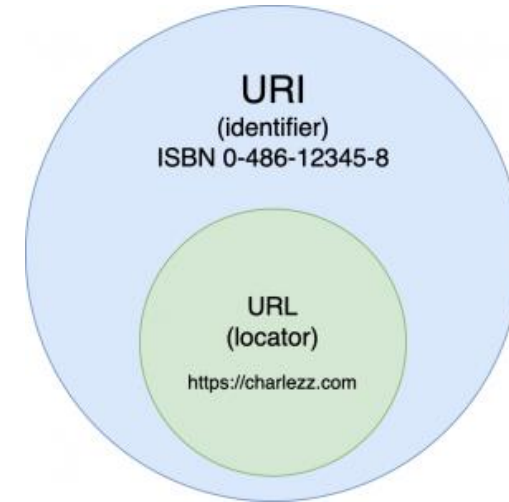
Deleter : DELETE (데이터 삭제)

REST는 아래의 3가지로 구성

1. 자원(Resource) : HTTP URI → URI? →
2. 행위 : HTTP Method
3. 내용(Representations) : HTTP Message Pay Load

특징

- Server Client 구조
- Sateless
- Cacheable
- Layered System
- Uniform Interface



장점

HTTP 인프라 그대로 사용 → 별도 인프라 구축 필요 X

모든 플랫폼에서 사용가능

메시지를 통해서 의미 쉽게 파악 가능

서버와 클라이언트 역할 명확

단점

Method가 제한적

URL보다 header 정보의 값을 처리해야 하므로 어렵다

가끔 구형 브라우저에서는 지원이 안되는 경우가 발생

자자.. 그래서 REST는 알겠고,, RESTful API는 ?

REST의 원리를 따르는 API !!!!!

```
1  from flask import Flask, request, json, jsonify
2
3  app = Flask(__name__)
4
5  @app.route("/test", methods=['POST'])
6  def test():
7      params = request.get_json()
8      print("받은 Json 데이터 ", params)
9
10     response = {
11         "result": "ok"
12     }
13
14     return jsonify(response)
15
16
17  if __name__ == "__main__":
18     app.run(debug=True, host='0.0.0.0', port=8080)
```

1. URI는 동사보다는 명사를, 대문자보다는 소문자를 사용하여야 한다.

Bad Example <http://khj93.com/Running/>

Good Example <http://khj93.com/run/>

2. 마지막에 슬래시 (/)를 포함하지 않는다.

Bad Example <http://khj93.com/test/>

Good Example <http://khj93.com/test>

3. 언더바 대신 하이픈을 사용한다.

Bad Example http://khj93.com/test_blog

Good Example <http://khj93.com/test-blog>

4. 파일확장자는 URI에 포함하지 않는다.

Bad Example <http://khj93.com/photo.jpg>

Good Example <http://khj93.com/photo>

5. 행위를 포함하지 않는다.

Bad Example <http://khj93.com/delete-post/1>

Good Example <http://khj93.com/post/1>

<https://gyoogle.dev/blog/computer-science/operating-system/Process%20vs%20Thread.html>

QnA