

## 최대 길이 연속수열

매개변수 `nums`에 정렬되지 않은 수열이 주어지면 수열의 원소로 만들 수 있는 가장 긴 연속된 수열의 길이를 구하여 반환하는 프로그램을 작성하세요.

연속된 수열이란 1씩 증가하는 수열을 의미합니다.

만약 `nums = [8, 1, 9, 3, 10, 2, 4, 0, 2, 3]` 이면 이 수열의 원소로 만들 수 있는 가장 긴 연속된 수열은 `[0, 1, 2, 3, 4]`입니다.

입출력 예:

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

제한사항:

- `nums`의 길이는 300,000을 넘지 않습니다.
- `nums[i]`의 값은 -1,000,000,000에서 1,000,000,000까지이며, 중복된 값도 있습니다.

입력예제 1번 설명:

만들 수 있는 가장 긴 연속수열은 `[0, 1, 2, 3, 4]`입니다.

## 겹쳐진 압축 해제

압축된 결과의 문자열이 주어지면 다시 원 상태로 압축을 해제하려고 합니다.

압축된 결과가 3(ab) 라고 주어지면 괄호안에 문자열이 3번 반복된 것을 압축했다는 의미입니다. 이걸 원상태로 해제한 것은 "ababab"입니다.

만약 2(ab)k3(bc) 를 압축해제 하면 "ababkbcbcbc"입니다. 2(ab)k3(bc)에서 k문자와 같이 반복횟수가 1인 경우는 숫자를 생략하고 압축되어 있습니다.

또한 겹쳐서 압축된 2(a2(b))을 압축해제하는 과정은 2(a2(b))-->2(abb)-->abbabb로 2(a2(b))를 압축해제한 결과는 abbabb입니다.

매개변수 s에 압축된 결과를 표현하는 문자열이 주어지면 원상태로 압축을 해제한 결과를 반환하는 프로그램을 작성하세요.

입출력 예:

s	answer
"3(a2(b))ef"	"abbabbabbef"
"2(ab)k3(bc)"	"ababkbcbcbc"
"2(ab3((cd)))"	"abcdcdcdabcdcdcd"
"2(2(ab)3(2(ac)))"	"ababacacacacacacab abacacacacacac"
"3(ab2(sg))"	"absgsgabsgsgabsgsg"

제한사항:

- 괄호안의 문자열의 반복횟수는 30을 넘지 않습니다.
- 압축을 해제한 문자열의 최종길이는 1,000을 넘지 않는다.
- 문자열 s의 알파벳은 소문자로만 구성됩니다.

## 현관문 출입순서

현수가 다니는 회사 건물 1층에는 보안을 위한 작은 현관문이 있습니다.

이 현관문은 동시에 2명 이상 출입이 불가능합니다. 1초에 한명씩만 출입을 할 수 있는 문입니다.

0번부터  $n-1$ 번의 직원번호를 갖고 있는 사원이 현관문을 출입을 하는데 아래와 같은 규칙으로 출입을 합니다.

- 1) 1초 전에 현관문을 사용한 적이 없으면 나가는 사원이 먼저 현관문을 이용합니다.
- 2) 1초 전에 나가는 사원이 현관문을 이용했다면 나가는 사원이 먼저 현관문을 이용합니다.
- 3) 1초 전에 들어오는 사원이 문을 이용했다면 들어오는 사원이 먼저 현관문을 이용합니다.
- 4) 같은 방향으로 가려고 하는 사람이 여러명이라면 그 중 직원번호가 가장 작은 사람이 우선 현관문을 이용합니다.

매개변수 arrival에 0번 직원부터 순서대로 현관문에 도착한 시간이 주어지고, state에 0번 직원부터 순서대로 들어가는 직원인지, 나가는 직원인지 알려준다면, 각 사원이 현관문을 사용하는 시간(초)를 직원번호순으로 배열에 담아 반환하는 프로그램을 작성하세요.

입출력 예:

arrival	state	answer
[0, 1, 1, 1, 2, 3, 8, 8]	[1, 0, 0, 1, 0, 0, 1, 0]	[0, 2, 3, 1, 4, 5, 8, 9]
[3, 3, 4, 5, 5, 5]	[1, 0, 1, 0, 1, 0]	[3, 6, 4, 7, 5, 8]
[2, 2, 2, 3, 4, 8, 8, 9, 10, 10]	[1, 0, 0, 0, 1, 1, 0, 1, 1, 0]	[2, 3, 4, 5, 6, 8, 11, 9, 10, 12]

제한사항:

- $1 \leq n \leq 100,000$
- $0 \leq \text{arrival}[i] \leq 200,000$ , arrival[i]는 i번 사원이 현관문에 도착한 시간입니다.
- $i < j$  이면  $\text{arrival}[i] \leq \text{arrival}[j]$ 를 만족합니다.
- $0 \leq \text{state}[i] \leq 1$ , 0은 들어오는 사원, 1은 나가는 사원을 뜻합니다.

입력예제 1번 설명 :

0초에 0번 사원이 나갑니다. 1초에 3번 사원이 나갑니다. 2초에 1번 사원이 들어갑니다.

3초에 2번 사원이 들어갑니다. 4초에 4번 사원이 들어갑니다. 5초에 5번 사원이 나갑니다.

6초에는 아무도 현관문에 없습니다. 7초에는 아무도 현관문에 없습니다.

8초에 6번 사원이 나갑니다. 9초에 7번 사원이 들어갑니다.

## 피부과

현수는 피부과를 운영합니다. 고객들이 레이저 시술을 받기위해 기다리는 대기실이 작다는 불만이 접수되어 대기실을 확장하려고 합니다. 그래서 현수는 대기실에서 동시에 대기하는 최대 인원수를 알고 싶습니다.

매개변수 laser에 각 레이저 종류별 시술을 마치는데 필요한 시간(분), enter에 각 고객의 피부과 방문시간과 해당 고객이 시술받을 레이저 종류 정보가 주어지면 시술을 받기 위해 대기실에서 동시에 기다리는 최대인원이 몇 명 발생하는지 그 명수를 반환하는 프로그램을 작성하세요. 레이저 시술이 끝난 고객은 바로 집으로 간다고 간주합니다.

현수는 한 번에 한 고객만 치료합니다. 치료순서는 피부과에 온 순서대로입니다. 고객이 피부과에 온 시간은 모두 다릅니다.

입출력 예:

laser	enter	answer
[30, 20, 25, 15]	["10:23 0", "10:40 3", "10:42 2", "10:52 3", "11:10 2"]	3
[30, 20, 25, 15]	["10:23 0", "10:40 3", "10:42 2", "10:52 3", "15:10 0", "15:20 3", "15:22 1", "15:23 0", "15:25 0"]	4
[30, 20, 25, 15]	["10:20 1", "10:40 1", "11:00 1", "11:20 1", "11:40 1"]	0

제한사항:

- enter의 길이는 100,000을 넘지 않습니다.
- enter 원소의 입력순은 시간순으로 주어집니다.
- enter[i]의 첫 번째 문자열은 i번 고객이 방문한 시간, 두 번째 문자열은 i번 고객이 시술받을 레이저 번호입니다. 시간은 HH:MM형태이며 09:00부터 20:00까지입니다.
- laser의 길이는 10을 넘지 않습니다.
- laser[i]의 값은 i번 레이저가 한 사람의 고객을 시술하는데 걸리는 시간(분)입니다.

입력예제 1번 설명 :

0번 고객은 기다리지 않고 바로 10:23분에 시술에 들어가 30분동안 시술하고 10:53에 끝납니다. 이 시간동안 1번, 2번, 3번 고객이 대기실에 기다리고 있습니다. 즉 10:52분에 대기실에서 3명이 기다리게 되고 이 명수가 동시에 대기실에서 기다리는 최대인원이 됩니다.

## CPU 스케줄링

0번부터  $n-1$ 번까지 작업번호가 있는  $n$ 개의 작업이 있습니다.

각 작업은 [호출시간, 실행시간]으로 정보가 표현됩니다. 예를 들어 어떤 작업의 정보가 [2, 3] 이면 0초부터 시작하여 흐르는 시간에서 2초에 작업호출되어 대기상태에 들어가며 자신의 차례가 되면 3초동안 실행된 뒤 종료됩니다.

cpu는 다음과 같은 규칙에 의해서 작업을 실행시킵니다.

- 1) 한 번에 한 개의 작업을 할 수 있고, 한 번 실행된 작업은 중간에 멈추지 않습니다.
- 2) 대기상태에 있는 작업이 많을 경우 그 중 실행시간이 가장 작은 작업을 먼저 처리하며, 실행시간이 같은 작업의 경우는 작업번호가 작은 것을 먼저 처리합니다.
- 3) cpu는 한 작업이 끝나면 바로 다른 작업을 할 수 있습니다. 만약 어떤 작업이 5초에 끝나면 5초에 바로 다른 작업을 할 수 있습니다.

매개변수 tasks에 0번 작업부터 순서대로 각 작업의 정보가 주어진다면 cpu가 처리하는 작업의 순서대로 작업번호를 배열에 담아 반환하는 프로그램을 작성하세요.

입출력 예:

tasks	answer
[[2, 3], [1, 2], [8, 2], [3, 1], [10, 2]]	[1, 3, 0, 2, 4]
[[5, 2], [7, 3], [1, 3], [1, 5], [2, 2], [1, 1]]	[5, 4, 2, 0, 1, 3]
[[1, 2], [2, 3], [1, 3], [3, 3], [8, 2], [1, 5], [2, 2], [1, 1]]	[7, 0, 6, 1, 4, 2, 3, 5]
[[999, 1000], [996, 1000], [998, 1000], [999, 7]]	[1, 3, 0, 2]

제한사항:

- tasks의 길이는 10,000 이하의 자연수입니다.
- 각 작업의 호출시간과 실행시간은 10,000을 넘지 않습니다.

입력예제 1번 설명 :

1초에 1번 작업을 실행해서 3초에 끝냅니다.

3초에 3번 작업을 실행해서 4초에 끝냅니다.

4초에 0번 작업을 실행해서 7초에 끝냅니다.

8초에 2번 작업을 실행해서 10초에 끝냅니다.

10초에 4번 작업을 실행해서 12초에 끝냅니다.

## 가장 많이 사용된 회의실

현수가 다니는 회사에서 여러개의 회의가 동시에 진행되는 행사를 기획하고 있습니다.

현수의 회사는 0번부터  $n-1$ 번으로 번호가 매겨진 회의실을  $n$ 개 가지고 있습니다.

각 회의는 다음과 같은 방식으로 회의실에 할당됩니다.

- 1) 사용 가능한 회의실이 생기면 시작 시간이 더 빠른 회의를 회의실에 배정합니다.
- 2) 사용 가능한 회의실이 여러개일 경우 가장 번호가 낮은 회의실에 회의를 배정합니다.
- 3) 사용 가능한 회의실이 없는 경우 다음 순서의 회의는 회의실이 비워질 때까지 기다렸다 자신의 회의 시간만큼 회의를 합니다. 만약 회의가 7시에 끝나고 회의실이 7시에 비워진다면 기다린 회의가 바로 7시에 시작될 수 있다고 가정합니다.

매개변수  $n$ 에 회의실의 개수가 주어지고, 매개변수 `meetings`에 각 회의의 시작시간과 끝나는 시간이 주어지면 가장 많은 회의를 개최한 회의실의 번호를 반환합니다. 답이 여러개일 경우 번호가 가장 작은 회의실의 번호를 반환합니다.

입출력 예

n	meetings	answer
2	[[0, 5], [2, 7], [4, 5], [7, 10], [9, 12]]	0
3	[[3, 9], [1, 10], [5, 8], [10, 15], [9, 14], [12, 14], [15, 20]]	0
3	[[1, 30], [2, 15], [3, 10], [4, 12], [6, 10]]	1
4	[[3, 20], [1, 25], [5, 8], [10, 15], [9, 14], [12, 14], [15, 20]]	2

제한사항:

- `meetings`의 길이는 200,000입니다. 각 회의에 시작시간은 고유합니다.
- $1 \leq n \leq 200$
- 회의시간은  $0 \leq \text{회의시간} \leq 500,000$ 입니다.

예제 1번 설명 :

[0, 5] 회의가 0번 회의실에 배정됩니다.

[2, 7] 회의가 1번 회의실에 배정됩니다.

[4, 5] 회의는 기다렸다 [0, 5] 회의가 끝나고 [5, 6]동안 회의를 0번 회의실에서 합니다.

[7, 10]회의는 0번, 1번 회의실 모두 사용할 수 있지만 이 경우 0번 회의실에 배정됩니다.

[9, 12]회의는 1번 회의실에 배정됩니다.

0번은 3개의 회의를 개최하고, 1번은 2개의 회의를 개최합니다.

답은 0번 회의실입니다.