

팰린드롬의 경우수

매개변수 `s`에 문자열이 주어지면 이 문자열의 문자들을 가지고 만들 수 있는 팰린드롬의 경우들을 배열에 담아 반환하는 프로그램을 작성하세요. 팰린드롬의 순서는 상관없습니다. 만약 팰린드롬이 만들어 지지 않을 경우 빈배열을 반환합니다.

입출력 예:

s	answer
"aaaabb"	['baaaab', 'aabbbaa', 'abaaba']
"abbcc"	["bcacb", "cbabc"]
"abbccce"	["bceaecb", "becaceb", "cbeaebc", "cebabec", "ebcacbe", "ecbabce"]
"abbcccee"	[]

제한사항:

- 문자열 `s`의 길이는 16를 넘지 않습니다.
- 문자열 `s`는 소문자로만 구성됩니다.

IP 주소

매개변수 `s`에 숫자로 구성된 문자열이 주어지면 이 문자열을 이용하여 유효한 IP주소의 경우들을 찾고 싶습니다.

만약 문자열 `s = "2025505"` 이 주어지면 만들 수 있는 유효한 IP 주소는 `["20.25.50.5", "20.255.0.5", "202.5.50.5", "202.55.0.5"]` 와 같습니다.

- 1) IP 주소는 4개의 숫자가 . 로 구분되어 있습니다.
- 2) IP 주소의 4개의 숫자는 0 ~ 255사이의 숫자로 구성됩니다.(0, 255포함)
- 3) IP 주소는 4개의 숫자는 0으로 시작해서는 안됩니다.

위에 입력의 경우 `"2.0.255.0.5"`와 같이 5개로 구성되어 있거나, `"2.025.50.5"`와 같이 0으로 시작하거나, `"20.2.5.505"`와 같이 255보다 큰 숫자가 존재하거나 하면 안됩니다.

매개변수 `s`에 숫자로 구성된 문자열이 주어지면 이 문자열을 이용하여 유효한 IP주소의 경우들을 배열에 담아 반환하는 프로그램을 작성하세요. 순서는 상관없습니다.

매개변수 `s`의 숫자들의 순서를 임의로 바꾸거나, 숫자를 지워서는 안됩니다.

만들 수 있는 유효한 IP가 없을 경우 빈 배열을 반환합니다.

입출력 예:

s	answer
"2025505"	["20.25.50.5", "20.255.0.5", "202.5.50.5", "202.55.0.5"]
"0000"	["0.0.0.0"]
"255003"	["25.50.0.3", "255.0.0.3"]
"155032012"	[]
"101023"	[1.0.10.23, 1.0.102.3, 10.1.0.23, 10.10.2.3, 101.0.2.3]

제한사항:

- 문자열 `s`의 길이는 16를 넘지 않습니다.
- 문자열 `s`는 숫자로만 구성되어 있습니다.

알파코드

철수와 영희는 서로의 비밀편지를 암호화해서 서로 주고받기로 했다.

그래서 서로 어떻게 암호화를 할 것인지 의논을 하고 있다.

영희 : 우리 알파벳 A에는 1로, B에는 2로 이렇게 해서 Z에는 26을 할당하여 번호로 보내기로 하자.

철수 : 정말 바보같은 생각이군!! 생각해 봐!! 만약 내가 "BEAN"을 너에게 보낸다면 그것을 암호화하면 25114이잖아!! 그러면 이것을 다시 알파벳으로 복원할 때는 많은 방법이 존재하는데 어떻게 할건데... 이것을 알파벳으로 바꾸면 BEAAD, YAAD, YAN, YKD 그리고 BEKD로 BEAN말고도 5가지나 더 있군.

당신은 위와 같은 영희의 방법으로 암호화된 코드가 주어지면 그것을 알파벳으로 복원하는데 얼마나 많은 방법이 있는지 알고 싶습니다.

매개변수 s에 암호화된 숫자가 문자열형태로 주어지면 이 것을 알파벳으로 복원하는 방법의 가지수를 반환하는 프로그램을 작성하세요.

입출력 예:

s	answer
"25114"	6
"23251232"	12
"21020132"	2
"21350"	0
"120225"	3
"232012521"	12

제한사항:

- 문자열 s의 길이는 100를 넘지 않습니다.
- 문자열 s는 숫자로만 구성되어 있습니다.

이동한 숫자

원소값이 정수인 nums 배열이 있습니다.

1. $i=1$ 로 초기화하고 정렬알고리즘이 시작됩니다.
2. $nums[0]$ 부터 $nums[i-1]$ 까지 이미 정렬되어 있습니다.
3. $j=i-1$ 부터 0까지 1씩 작아지면서 $nums[j]$ 값이 $nums[i]$ 보다 크면 $nums[j]$ 는 자신의 뒷 칸으로 이동시키고 $nums[j]$ 가 $nums[i]$ 보다 작거나 같으면 j 는 반복을 멈춥니다.
4. $nums[i]$ 값을 $j+1$ 위치에 저장합니다.
5. i 값을 1 증가시킵니다.
6. 2~5과정을 반복하면서 오름차순 정렬을 합니다.

회전	결과	이동 숫자	뒤로 이동한 숫자 개수
$i = 1$	[5, 4, 2, 1, 3]	4는 5를 뒤로 이동	1
$i = 2$	[4, 5, 2, 1, 3]	2는 4, 5를 뒤로 이동	2
$i = 3$	[2, 4, 5, 1, 3]	1은 2, 4, 5를 뒤로 이동	3
$i = 4$	[1, 2, 4, 5, 3]	3은 4, 5를 뒤로 이동	2
	[1, 2, 3, 4, 5]		

매개변수 nums에 수열이 주어지면 위와 같은 삽입정렬을 할 경우 두 번째 숫자부터 자신의 앞 숫자 몇 개를 뒤로 이동시키는지 배열에 저장해 반환하는 프로그램을 작성하세요.

단 여러분이 작성해야 할 프로그램의 시간복잡도는 $O(n \log n)$ 으로 해야 합니다.

힌트를 드리면 시간복잡도가 $O(n \log n)$ 인 병합정렬을 이용해서 배열에서 자신의 앞쪽(왼쪽)에 자신보다 큰 수가 몇 개 있는지 알 수 있습니다.

참고로 위에서 설명한 삽입정렬은 $O(n^2)$ 입니다.

입출력 예

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

제한사항

- nums의 길이는 200,000을 넘지 않습니다.

스도쿠(DFS 활용)

스도쿠는 9×9 크기의 보드가 있을 때, 각 행과 각 열, 그리고 9개의 3×3 크기의 보드에 1부터 9까지의 숫자가 중복 없이 나타나도록 보드를 채우는 퍼즐이다.

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

위 그림은 스도쿠를 정확하게 풀어낸 것이다. 각 행에 1부터 9까지의 숫자가 중복 없이 나오고, 각 열에 1부터 9까지의 숫자가 중복 없이 나오고, 각 3×3짜리 사각형(9개이며, 위에서 색깔로 표시되었다)에 1부터 9까지의 숫자가 중복 없이 한 번씩 나오기 때문이다.

매개변수 board에 완성되지 않은 스도쿠 퍼즐이 주어졌을 때, 스도쿠를 정확하게 완성해서 배열로 반환하는 프로그램을 작성하세요.

답이 여러 개 있다면 그 중 오름차순으로 제일 먼저 발견된 것을 반환하면 됩니다. 즉, 81자리의 수가 제일 작은 경우로 반환하는 것입니다.

입출력 예:

board	answer
[[0, 2, 3, 0, 5, 0, 7, 8, 9], [0, 5, 6, 0, 8, 9, 1, 0, 3], [0, 8, 9, 1, 0, 3, 0, 5, 6], [0, 1, 0, 0, 6, 0, 8, 9, 0], [3, 0, 5, 0, 9, 7, 0, 1, 4], [0, 9, 7, 0, 1, 0, 0, 6, 5], [5, 3, 0, 6, 0, 2, 9, 7, 8], [6, 0, 2, 9, 0, 8, 5, 3, 1], [9, 0, 8, 0, 3, 0, 6, 0, 2]]	[[1, 2, 3, 4, 5, 6, 7, 8, 9], [4, 5, 6, 7, 8, 9, 1, 2, 3], [7, 8, 9, 1, 2, 3, 4, 5, 6], [2, 1, 4, 3, 6, 5, 8, 9, 7], [3, 6, 5, 8, 9, 7, 2, 1, 4], [8, 9, 7, 2, 1, 4, 3, 6, 5], [5, 3, 1, 6, 4, 2, 9, 7, 8], [6, 4, 2, 9, 7, 8, 5, 3, 1], [9, 7, 8, 5, 3, 1, 6, 4, 2]]

제한사항:

- board에 아직 숫자가 채워지지 않은 칸은 0으로 표시됩니다.