

+ New chat

LeetCode: Gas S

New chat

New chat

Clear conversations

Upgrade to Plus

NEW

Dark mode

Updates & FAQ

Log out

< 3 / 3 >



Задача "Split Array into Consecutive Subsequences" с LeetCode заключается в том, чтобы определить, можно ли разделить данную массив на одно или более подпоследовательностей, где каждая подпоследовательность состоит из трех или более последовательных чисел.



Для решения этой задачи мы можем использовать словарь (dictionary) для хранения текущего количества концов каждой подпоследовательности, которые заканчиваются на предыдущем элементе массива. Мы будем обновлять этот словарь при прохождении по массиву. Когда мы встречаем элемент  $i$ , мы проверяем, есть ли подпоследовательность, которая заканчивается на  $i - 1$ . Если такая подпоследовательность есть, то мы добавляем  $i$  к этой подпоследовательности. Если же такой подпоследовательности нет, то мы создаем новую подпоследовательность, заканчивающуюся на  $i$ .

Кроме того, мы также должны отслеживать количество оставшихся чисел, которые могут быть добавлены к существующим подпоследовательностям. Если мы не можем добавить число к существующей подпоследовательности, то мы должны создать новую подпоследовательность, начиная с этого числа.

Вот код на Python, реализующий описанный алгоритм:

python

Copy code

```
def isPossible(nums):
    ends = {}
    for num in nums:
        if num in ends:
            ends[num] -= 1
            if ends[num] == 0:
                del ends[num]
            if num + 1 in ends:
                ends[num + 1] += 1
            else:
                ends[num + 1] = 1
        elif num + 1 in nums and num + 2 in nums:
            nums.remove(num + 1)
            nums.remove(num + 2)
            if num + 3 in ends:
                ends[num + 3] += 1
            else:
                ends[num + 3] = 1
        else:
            return False
    return True
```

В этом коде мы используем словарь `ends` для отслеживания концов каждой подпоследовательности. Если число уже является концом подпоследовательности, то мы добавляем единицу к соответствующему значению в словаре `ends`. Если число не является концом подпоследовательности, то мы проверяем, есть ли следующие два числа в массиве. Если следующие два числа есть, то мы создаем новую подпоследовательность, начинающуюся с текущего числа, и добавляем ее конец в словарь `ends`. Если следующих двух чисел нет, то мы не можем создать новую подпоследовательность, и поэтому возвращаем `False`.

Н

Regenerate response

