**347. Top K Frequent Elements**

Medium

1837114FavoriteShare

Given a non-empty array of integers, return the ***k*** most frequent elements.

**Example 1:**

**Input:** nums = [1,1,1,2,2,3], k = 2

**Output:** [1,2]

**Example 2:**

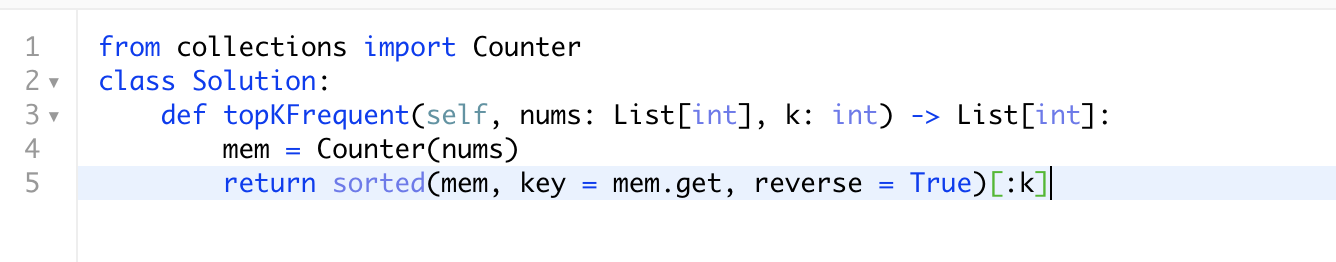
**Input:** nums = [1], k = 1

**Output:** [1]

**Note:**

* You may assume *k* is always valid, 1 ≤ *k* ≤ number of unique elements.
* Your algorithm's time complexity **must be** better than O(*n* log *n*), where *n* is the array's size.

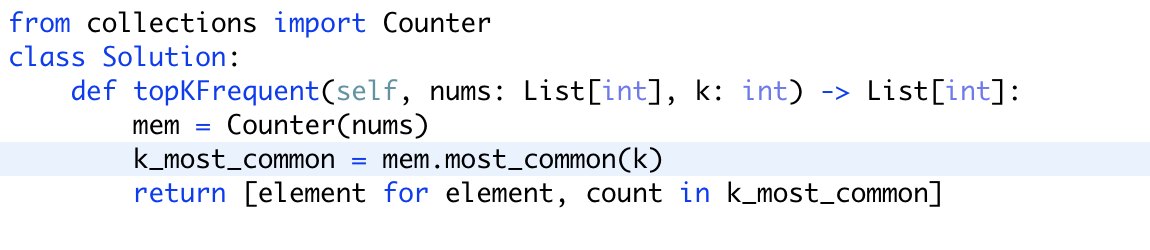
1. Counter + sorted



<https://docs.python.org/zh-cn/3/library/heapq.html>

key -- 主要是用来进行比较的元素，只有一个参数，具体的函数的参数就是取自于可迭代对象中，指定可迭代对象中的一个元素来进行排序。

1. Counter + most\_common



<https://docs.python.org/zh-cn/3/library/collections.html#collections.Counter>

使用counter自带的功能

**692. Top K Frequent Words**

Medium

90985FavoriteShare

Given a non-empty list of words, return the *k* most frequent elements.

Your answer should be sorted by frequency from highest to lowest. If two words have the same frequency, then the word with the lower alphabetical order comes first.

**Example 1:**

**Input:** ["i", "love", "leetcode", "i", "love", "coding"], k = 2

**Output:** ["i", "love"]

**Explanation:** "i" and "love" are the two most frequent words.

Note that "i" comes before "love" due to a lower alphabetical order.

**Example 2:**

**Input:** ["the", "day", "is", "sunny", "the", "the", "the", "sunny", "is", "is"], k = 4

**Output:** ["the", "is", "sunny", "day"]

**Explanation:** "the", "is", "sunny" and "day" are the four most frequent words,

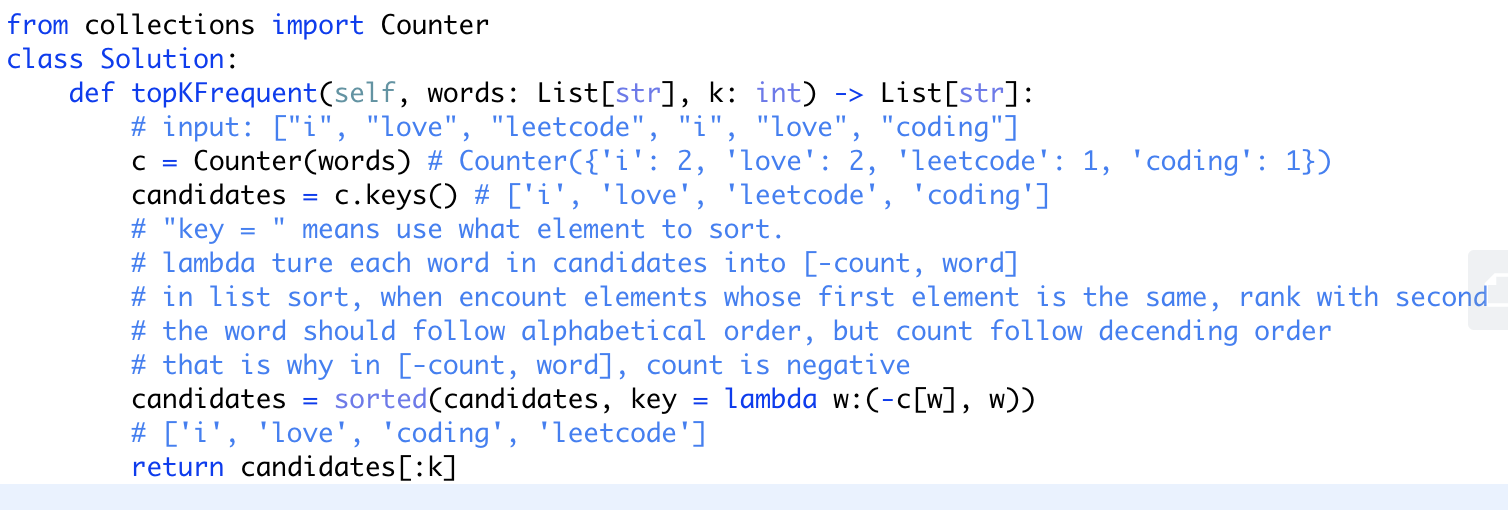
with the number of occurrence being 4, 3, 2 and 1 respectively.

**Note:**

1. You may assume *k* is always valid, 1 ≤ *k* ≤ number of unique elements.
2. Input words contain only lowercase letters.

**Follow up:**

1. Try to solve it in *O*(*n* log *k*) time and *O*(*n*) extra space.



<https://www.cnblogs.com/evening/archive/2012/03/29/2423554.html>

