

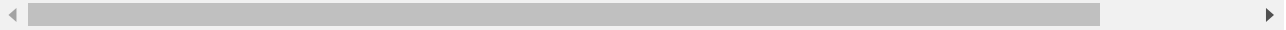
Sort the given iterable so that its elements end up in the decreasing frequency order, that is, the number of times they appear in elements. If two elements have the same frequency, they should end up in the same order as the first appearance in the iterable.

Input: Iterable

Output: Iterable

Example:

```
1 frequency_sort([4, 6, 2, 2, 6, 4, 4, 4]) == [4, 4, 4, 4, 6, 6, 2, 2]
2 frequency_sort(['bob', 'bob', 'carl', 'alex', 'bob']) == ['bob', 'bob', 'bob', 'ca
```



Precondition: *elements can be ints or strings*

The mission was taken from Python CCPS 109 Fall 2018. It's being taught for Ryerson Chang School of Continuing Education by [Ilkka Kokkarinen](#)