



# [CS 11] Prac 10h – Warehouse Sort II

## Problem Statement

Several packages have just arrived at the Chopee warehouse! Specifically, a total of  $n$  packages arrived. We can index them from 0 to  $n - 1$ .

Each package is labelled with an ID, which is a string consisting of letters and digits. The label of package  $i$  is  $\ell_i$ .

Now, your task is to sort the packages by label. However, you're not sure whether the comparison should be case-sensitive or not, so you want to account for both possibilities. Also, if there are ties, you must break ties according to the package's index.

Given the IDs of the  $n$  packages, please return the package indices in the correct sorted order.

**Note:** When comparing strings lexicographically, spaces come before letters. Also, when comparing case-sensitively, uppercase letters come before lowercase letters.

## Task Details

Your task is to implement a function called `sorted_packages`. This function has a single positional argument, a `tuple`/`list` of  $n$  `str`s representing the labels of the  $n$  packages:  $\ell_0$  through  $\ell_{n-1}$ . It also has a single keyword argument, `case_sensitive`, a `bool` denoting whether the comparison should be case-sensitive or not. It should default to `True`.

The function must return a `list` of  $n$  `int`s denoting the arrangement of the indices 0 to  $n - 1$  ordered by label (breaking ties by index). More precisely, if the returned sequence is  $i_0, i_1, \dots, i_{n-1}$ , then the first package is package  $i_0$ , then the next one is package  $i_1$ , then the next one is package  $i_2$ , etc.

## Restrictions

(See 10a for more restrictions)

For this problem in particular:

- The following symbols are allowed: `map`, `filter`.
- The following import is allowed: `cache` and `lru_cache` from `functools`.
- The source code limit is 1000.

## Example Calls

### Example 1 Function Call

```
sorted_packages([
    'iPhone',
    'iPHONE',
    'USB converter',
    'Flash Drive',
    'J Ball speaker',
    'uPhone',
    'cooking pot',
    'headset cup holder',
    'insect racket',
])
```

Copy

### Example 1 Return Value

```
[3, 4, 2, 6, 7, 1, 0, 8, 5]
```

Copy

### Example 2 Function Call

```
sorted_packages([
    'iPhone',
    'iPHONE',
    'USB converter',
    'Flash Drive',
    'J Ball speaker',
    'uPhone',
    'cooking pot',
    'headset cup holder',
    'insect racket',
], case_sensitive=False)
```

Copy

### Example 2 Return Value

```
[6, 3, 7, 8, 0, 1, 4, 5, 2]
```

Copy

## Constraints

- The function `sorted_packages` will be called at most 60,000 times.
- The sum of  $n$ s across all calls will be  $\leq 250,000$ .
- $0 \leq n \leq 250,000$
- Each label will be a string with at most 18 English letters or spaces.

## Scoring

- You get 60 ❤️ points if you solve all test cases where:
  - $n \leq 50$
  - The sum of the  $n$ s across all calls is 500.
  - The `case_sensitive` keyword argument is not used/passed.
- You get 30 ❤️ points if you solve all test cases where:
  - $n \leq 50$
  - The sum of the  $n$ s across all calls is 500.
- You get 35 ❤️ points if you solve all test cases where:
  - $n \leq 4,000$
  - The sum of the  $n$ s across all calls is 8,000.
- You get 55 ❤️ points if you solve all test cases.

## ❓ Clarifications

Report an issue

No clarifications have been made at this time.