[CS 11] Prac 8d – Geegle Search

Submit solution [CS 11] Practice 8 ❤

My submissions ❤

✓ **Points:** 180 (partial)
Time limit: 4.0s
Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Problem Statement

You have just been hired at Geegle, Inc.!

They tasked you with developing a search functionality in their online store. There are several product names, each with its own price.

Several queries have come in! Each query consists of a product name. Can you determine the price of each of those products?

If there is no product with that name, respond with "not found" instead. This means that even a small typo means the search might fail!

However, the search must be **case-insensitive**. For example, a product name of `QuiwaBook` should be searchable with keyword `quiwabook`.

Task Details

Your task is to implement a function called `search_prices`. This function has two parameters:

- the first is a `tuple` of pairs representing the products and their prices. Each pair consists of:
  - a `str` representing the product name.
  - an `int` representing the price.
- the second is a `tuple` of `str`s representing the query texts.

The function must return a `list` of exactly the same length as the other argument, representing the responses to each query. Each response is either an `int` representing the price, or the string `"not found"` if a product with that name doesn't exist.

Restrictions

(See 8a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 450.

Example Calls

Example 1 Function Call

Copy

```
search_prices((
    ('Porkchop', 80),
    ('Kanin', 15),
    ('Ketchup', 50),
    ('ChopSuey', 60),
    ('Bulalo', 100),
    ('Adobo', 70),
    ('Chicken', 70),
    ('Mami', 50),
    ('Longsilog', 100),
    ('Tocilog', 120),
    ('Tapsi', 80),
    ('Pancit', 100),
    ('Lugaw', 40),
    ('Lomi', 60),
    ('TokwatBaboy', 40),
    ('ParesBeef', 40),
    ('FriedSiomai', 35),
    ('Siopao', 60),
), (
    'Kanin',
    'KANIN',
    'LongSiLog',
    'chicKen',
    'adobe',
    'xiaomai',
    'Mami',
    'Dadi',
    'Bebi',
))
```

Example 1 Return Value

Copy

```
[
    15,
    15,
    100,
    70,
    "not found",
    "not found",
    50,
    "not found",
    "not found",
]
```

Constraints

- The function `search_prices` will be called at most $60{,}000$ times.
- The total number of products across all calls will be at most $200{,}000$.
- The total number of searches across all calls will be at most $200{,}000$.
- The number of products in each call will be at most $200{,}000$.
- The number of searches in each call will be at most $200{,}000$.
- Each price is an integer between 1 and 10^{10}.
- Each word or search text is a nonempty string of at most 12 English letters.
- The product names are distinct, even ignoring case.

Scoring

- You get 30 ❤ points if you solve all test cases where:
  - the number of products in each call is ≤ 50.
  - the number of searches in each call is ≤ 50.
  - the total number of products across all calls is at most 500.
  - the total number of searches across all calls is at most 500.
- You get 30 ❤ points if you solve all test cases where:
  - the number of products in each call is $\leq 4{,}000$.
  - the number of searches in each call is $\leq 4{,}000$.
  - the total number of products across all calls is at most $8{,}000$.
  - the total number of searches across all calls is at most $8{,}000$.
- You get 120 ❤ points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.