Problem 2 - Dealing with Hands

Problem Set due Jul 16, 2020 20:30 -03 *Completo*

Problem 2 - Dealing with Hands

10/10 points (graded)

****Please read this problem entirely!!**** The majority of this problem consists of learning how to read code, which is an incredibly useful and important skill. At the end, you will implement a short function. Be sure to take your time on this problem - it may seem easy, but reading someone else's code can be challenging and this is an important exercise.

Representing hands

A **hand** is the set of letters held by a player during the game. The player is initially dealt a set of random letters. For example, the player could start out with the following hand: `a, q, l, m, u, i, l`. In our program, a hand will be represented as a dictionary: the keys are (lowercase) letters and the values are the number of times the particular letter is repeated in that hand. For example, the above hand would be represented as:

```
hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
```

Notice how the repeated letter `'l'` is represented. Remember that with a dictionary, the usual way to access a value is `hand['a']`, where `'a'` is the key we want to find. However, this only works if the key is in the dictionary; otherwise, we get a `KeyError`. To avoid this, we can use the call `hand.get('a',0)`. This is the "safe" way to access a value if we are not sure the key is in the dictionary. `d.get(key,default)` returns the value for `key` if `key` is in the dictionary `d`, else `default`. If `default` is not given, it returns `None`, so that this method never raises a `KeyError`. For example:

```
>>> hand['e']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'e'
>>> hand.get('e', 0)
0
```

Converting words into dictionary representation

One useful function we've defined for you is `getFrequencyDict`, defined near the top of `ps4a.py`. When given a string of letters as an input, it returns a dictionary where the keys are letters and the values are the number of times that letter is represented in the input string. For example:

```
>>> getFrequencyDict("hello")
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

As you can see, this is the same kind of dictionary we use to represent hands.

Displaying a hand

Given a hand represented as a dictionary, we want to display it in a user-friendly way. We have provided the implementation for this in the `displayHand` function. Take a few minutes right now to read through this function carefully and understand what it does and how it works.

Generating a random hand

The hand a player is dealt is a set of letters chosen at random. We provide you with the implementation of a function that generates this random hand, `dealHand`. The function takes as input a positive integer `n`, and returns a new object, a hand containing `n` lowercase letters. Again, take a few minutes (right now!) to read through this function carefully and understand what it does and how it works.

Removing letters from a hand (you implement this)

The player starts with a hand, a set of letters. As the player spells out words, letters from this set are used up. For example, the player could start out with the following hand: **`a, q, l, m, u, i, l`**. The player could choose to spell the word **`quail`** . This would leave the following letters in the player's hand: **`l, m`**. Your task is to implement the function `updateHand`, which takes in two inputs - a `hand` and a `word` (string). `updateHand` uses letters from the hand to spell the word, and then returns a copy of the `hand`, containing only the letters remaining. For example:

```
>>> hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
>>> displayHand(hand) # Implemented for you
a q l l m u i
>>> hand = updateHand(hand, 'quail') # You implement this function!
>>> hand
{'a':0, 'q':0, 'l':1, 'm':1, 'u':0, 'i':0}
>>> displayHand(hand)
l m
```

Implement the `updateHand` function. Make sure this function has no side effects: i.e., it must not mutate the hand passed in. Before pasting your function definition here, be sure you've passed the appropriate tests in `test_ps4a.py`.

Hints

Testing

Copying Dictionaries

Your implementation of updateHand should be short (ours is 4 lines of code). It does not need to call any helper functions.

```
def updateHand(hand, word):
    """
    Assumes that 'hand' has all the letters in word.
    In other words, this assumes that however many times
    a letter appears in 'word', 'hand' has at least as
    many of that letter in it.

    Updates the hand: uses up the letters in the given word
    and returns the new hand, without those letters in it.

    Has no side effects: does not modify hand.

    word: string
    hand: dictionary (string -> int)
    returns: dictionary (string -> int)
    """
```

Press ESC then TAB or click outside of the code editor to exit

Correta

Test results

Hide output

CORRECT

Test 1

Function call: updateHand({'q': 1, 'a': 1, 'l': 2, 'u': 1, 'i': 1, 'm': 1}, 'quail')

Output:

```
{'l': 1, 'a': 0, 'u': 0, 'q': 0, 'm': 1, 'i': 0}
```

Test 2

Function call: updateHand({'r': 2, 'a': 2, 'p': 3, 'l': 2, 'c': 2, 't': 2}, 'claptrap')

Output:

```
{'c': 1, 'l': 1, 'a': 0, 'p': 1, 't': 1, 'r': 1}
```

Test 3

Function call: updateHand({'o': 1, 'g': 1, 'd': 1}, 'dog')

Output:

```
{'g': 0, 'o': 0, 'd': 0}
```

Test 4

Re-testing last test to see if you mutate the original hand

Output:

```
{'g': 0, 'o': 0, 'd': 0}
```

Test 4

Function call: updateHand({'o': 3, 'q': 3, 'w': 3, 'r': 3, 'y': 3, 'p': 3, 'u': 3, 'i': 3, 'e': 3, 't': 3}, 'typewriter')

Output:

```
{'r': 1, 'u': 3, 'o': 3, 'y': 2, 'p': 2, 'i': 2, 'w': 2, 'q':
3, 'e': 1, 't': 1}
```

Random Test 1

Function call: updateHand({'y': 1, 'b': 1, 'u': 1, 'x': 1, 'e': 2, 'i': 1, 'h': 1, 't': 2}, 'teeth')

Output:

```
{'x': 1, 'u': 1, 'y': 1, 'i': 1, 'b': 1, 'e': 0, 'h': 0, 't':
0}
```

Random Test 2

Re-testing last test to see if you mutate the original hand

Output:

```
{'x': 1, 'u': 1, 'y': 1, 'i': 1, 'b': 1, 'e': 0, 'h': 0, 't':
0}
```

Random Test 3

Function call: updateHand({'v': 1, 'd': 1, 'c': 3, 'u': 1, 'p': 2, 'k': 1}, 'duck')

Output:

```
{'c': 2, 'v': 1, 'u': 0, 'd': 0, 'p': 2, 'k': 0}
```

Random Test 4

Re-testing last test to see if you mutate the original hand

Output:

```
{'c': 2, 'v': 1, 'u': 0, 'd': 0, 'p': 2, 'k': 0}
```

Random Test 5

Function call: updateHand({'s': 2, 'o': 1, 'r': 1, 'a': 1, 'c': 1, 'e': 2, 'h': 1}, 'shoe')

Output:

```
{'c': 1, 'a': 1, 's': 1, 'o': 0, 'e': 1, 'h': 0, 'r': 1}
```

```
Random Test 6
```

Re-testing last test to see if you mutate the original hand

Output:

```
{'c': 1, 'a': 1, 's': 1, 'o': 0, 'e': 1, 'h': 0, 'r': 1}
```

```
Random Test 7
```

Function call: updateHand({'s': 1, 'o': 1, 'g': 1, 'r': 1, 'a': 1, 'b': 2, 'i': 1, 'j': 1}, 'boar')

Output:

```
{'g': 1, 'r': 0, 'o': 0, 's': 1, 'i': 1, 'j': 1, 'b': 1, 'a':
0}
```

```
Random Test 8
```

Re-testing last test to see if you mutate the original hand

Output:

```
{'g': 1, 'r': 0, 'o': 0, 's': 1, 'i': 1, 'j': 1, 'b': 1, 'a':
0}
```

Hide output

Enviar

You have used 2 of 30 attempts

✓ Correct (10/10 points)

Problem 2 - Dealing with Hands

Ocultar discussão

Topic: Problem Set 4 / Problem 2

Show all posts por atividade recente

? How is the order of the letters in hand are rearranged?

? absent return statement
The displayHand function does not have a return statement at the end. Anyone caring to elaborate o... 4

issue with dealHand
If you read the code for dealHand, the number of vowels will be exactly n // 3 each time and not at le... 3

Used getFrequencyDict() in my solution
Hi, in my solution to do this, I used the getFrequencyDict() helper code that was given to us to use in ... 2