Computer Systems Modeling and Verification (USEEN1)

Dictionaries, Recursion

Exercises taken from the *Python Programming Primer*[1] (and updated with type hints).

Textbook *Introduction to Python for Computational Science and Engineering* (2022), Chapter 3.

Corresponding lecture slides from *Computational Science and Engineering in Python*:

- Dictionaries
- Recursion

Tutorial. Define the following functions.

1. A function `list_to_dict[K, V](l: list[tuple[K, V]]) -> dict[K, V]` which converts a list of pairs into a dictionary.

2. A function `dict_to_list[K, V](d: dict[K, V]) -> list[tuple[K, V]]` which converts a dictionary into a list of pairs.

Exercices. Define the following functions.

1. A function `count_chars(s: str) -> dict[str, int]` which takes a string `s` and returns a dictionary. The dictionary's keys are the set of characters that occur in string `s`. The value for each key is the number of times that this character occurs in the string `s`.

   Examples:

   ```
   >>> count_chars("x")
   {'x': 1}

   >>> count_chars("xxx")
   {'x': 3}

   >>> count_chars("xxxyz")
   {'x': 3, 'y': 1, 'z': 1}

   >>> count_chars("Hello␣World")
   {'␣': 1, 'H': 1, 'W': 1, 'd': 1, 'e': 1, 'l': 3, 'o': 2, 'r': 1}
   ```

   Note that the order in which the key-value pairs are listed in the output dictionary is not important.

2. A function `is_palindrome(s: str) -> bool` which takes a string `s` and returns the value `True` if `s` is a palindrome, and returns `False` otherwise.

   A palindrome is a word that reads the same backwards as forwards, such as *madam*, *kayak*, *radar* and *rotator*.

   Hints for a suggested algorithm:

   - if `s` is an empty string, then it is a palindrome.
   - if `s` is a string with one character, then it is a palindrome.

[1] *Python Programming Primer*, Hans Fangohr *et al.* University of Southampton (2016)

- if the first letter of `s` is the same as the last letter of `s`, then `s` is a palindrome if the remaining letters of `s` (i.e. starting from the second letter, excluding the last letter) are a palindrome.

Examples:

```
>>> is_palindrome("rotator")
True

>>> is_palindrome("radiator")
False

>>> is_palindrome("ABBA")
True
```

We treat small letters (e.g. `a`) and capital letters (e.g. `A`) as different letters for this exercise: the string `ABba` is thus *not* a palindrome.

Suggestion: if you struggle with the concept of recursion, take some time to study the output of this recursive factorial computation.