ADVANCED PYTHON HOMEWORK 2

Voting, Recursing, Compressing

Every exercise is worth 2 points, and only 3 submissions are graded (so there is no need to submit more). This homework sheet is equivalent to the Polish one; if you do not understand anything, feel free to ask your lab teacher or consult the Polish sheet.

Exercise 1. The seats in a Polish parliamentary election or a Polish local government election are divided between the individual political parties or election committees by the *d'Hondt method*. Implement a function that takes the *election results* data structure as arguments, then a number of seats to fill, and returns a list of elected candidates.

We assume that

- the method works the same as described in: `https://en.wikipedia.org/wiki/D%27Hondt_method`.
- the election threshold is 5% (we do not include minority election committees, national or ethnic groups for which thresholds are waived).

Election results is a data structure storing a number of votes for each candidate including the election committees to which each candidate belongs. A good test for your solution is looking at Polish election results e.g. at `https://pkw.gov.pl/`, (`https://wybory2018.pkw.gov.pl/`) and check if the distribution of seats obtained by your algorithm is the same as the one provided by National Electoral Commission (Państwowa Komisja Wyborcza).

Exercise 2. Using the following formula:

$$\sum_{i=1}^{k} 2i - 1 = k^2$$

implement a function `square_root(n)` that computes $\lfloor \sqrt{n} \rfloor$.

Exercise 3. Implement a function `sudan(n, x, y)` defined by the recurrence relation:

$$\begin{aligned} F_0(x, y) &= x + y \\ F_{n+1}(x, 0) &= x, x \geq 0 \\ F_{n+1}(x, y + 1) &= F_n(F_{n+1}(x, y), F_{n+1}(x, y) + y + 1) \end{aligned}$$

The values of this function grow rapidly, so, you should be careful and avoid testing your programs for $n > 2$. To make the evaluation faster, your program should store the results that have been already calculated; such technique is called *memoization*.

Experimentally find the largest arguments for which is reasonable to call the function in the version with memoization and without memoization. Write your results into the documentation/comments.

Exercise 4. Sentences which are too long can be a nuisance for the reader. *(Translator's note: Why waste time say lot word when few word do trick?)* Because of that, we will simplify the text as follows:

- Delete all words which are too long.
- Next, remove words at random if the sentence has too many words still.

Your goal is to implement a function `simplify_sentence(text, word_length, word_number)`, where `word_length` is a maximal allowed word length and `word_number` is a maximal number of words in the sentence. For example:

```
tekst = "Podział peryklinalny inicjałów wrzecionowatych \
kambium charakteryzuje się ścianą podziałową inicjowaną \
```

```
                w płaszczyźnie maksymalnej."

simplify_sentence(text, 10, 5)
```

The simplified sentence may look something like this:

```
"Podział kambium się ścianą inicjowaną."
```

Test your program on some popular literary work that is legally available online. Please include a code that downloads the chosen text or add a link to it in the documentation/comments.

Exercise 5. One of the simple algorithms for text compression is to substitute a consecutive occurrences of the same character for a pair `(character, frequency)`; for example: instead of a string 'aaaaaa' [1] we can write `{['a', 5]}`. Text: 'suuuuper' is compressed to: `[(1, 's'), (4, 'u'), (1, 'p'), (1, 'e'), (1, 'r')]`. Your task is to implement two functions: `compression(text)` and `decompression(compressed_text)`, which return the compressed text and decompressed text, respectively. You can assume that we only compress texts that contain letters and punctuation marks (including spacing).

Test your program on some popular literary work that is legally available online. Please include a code that downloads the chosen text or add a link to it in the documentation/comments.

[1] Translator's note: Original text has 'aaaaaa' here. I will let you know via MS Teams if this was a typo.