

Python Programming

Exercise 2: Lists and strings

Prof. Dr. Thomas Kopinski

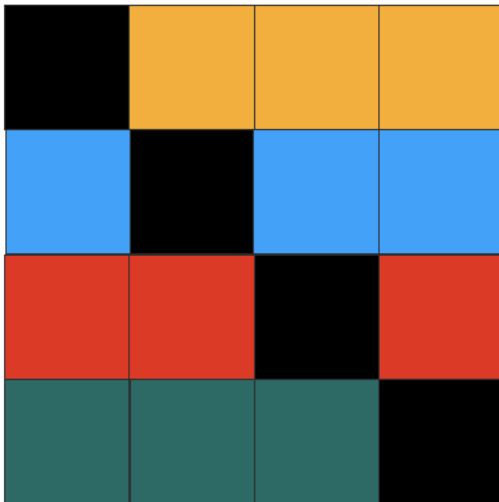
October 19, 2021

Abstract

This exercise will focus on the main data structure - lists - and operations with strings.

Task 1: Working with lists

Matrices play an important role in mathematics and therefore also in machine learning. Matrix A with entries a_{mn} is depicted as in Fig. 1b



(a) Matrix Schema

$$\begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{matrix} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{matrix}$$

(b) Matrix 2

Entries in a matrix follow a specific pattern - the row is referenced with the first index m , each column is referenced with the n index and in the case of entries on the main diagonal, $m = n$.

Write a Python program with the following characteristics:

- Do not use any external packages
- Create a data structure using lists capable of holding multiple entries with integers along any axes
- In our example, matrices will be of squared size i.e. the number of rows equals the number of columns
- For a machine learning program to test its performance it needs to compare its good decisions vs. its bad decisions. Good decisions are entries on the main diagonal with $m = n$, bad decisions are other entries with $m \neq n$. More specifically, we need to compare these numbers row-wise e.g. a_{00} vs. $a_{01} + a_{02} + \dots + a_{0n}$. Fig. 1a depicts this behavior where each black box in a row is compared to all other non-black boxes of the same color.

- Write a Python program which, for any given squared matrix with integer entries calculates for all i row-wise the quotient of $\frac{a_{ii}}{\sum_{j=0}^n a_{ij}}$
- Test your program with various matrices of different sizes

Task 2: Lists and strings

- Write a function in Python considering the following parameters: The function is given as parameters two integers a, b and a string s . The values of a and b determine the starting and ending of a substring $substr$ within s . For a given list of strings named *string_list* it tests whether *substr* is in any element of *string_list*
- Change the function of the task above so that it accepts also the possibility of finding the reverse of *substr* within *string_list*
- Write another function with similar behavior so that it tests whether the reversed ending of s is the start of any string in *string_list*
- Write a function in Python checking whether any two concatenated strings $s1, s2$ in a list are similar to any other string $s3$ in the *string_list*
- Test all functions with various cases