Report on solution and implementation of Track-4 task

1) Problem description

The task requires to create an application visualizing 4D tensor (uint8 20x20x20x20) in any user-friendly form. High values should correspond to bright pixels, whereas low values to dark ones. Tensor must be visualized via 2D or 3D projections. Task must be implemented on either python or C++. Any 3rd libraries, used in the program, must be explained.

2) Approach explanation

Python language was chosen for implementing the task due to its easy syntax and comfortable raster graphics libraries. Additionally used modules include *matplotlib.pyplot* and *random*.

2.1) matplotlib.pyplot

Technical documentation on this module, which can be freely accessed via Internet, makes it pretty easy to learn and to get familiar with. The module also provides multiple options for picture customization which is why it was considered the best choice.

Tensor is displayed as 400x400 matrix. It is divided int 400 20x20 squares, each of those squares represents 2D projection of the tensor (if it is visualized as a 20x0 matrix of 20x20 matrices). Pixels range in color from dark purple (correspond to 0 value) to light yellow (correspond to 255 value). Title text and a diagram are added to make user more familiar with application's functionality.

2.2) random

This module was used to more clearly showcase the results of the application performance. It helps create a 400x400 matrix, consisting of random integers, ranging from 0 to 255, which is a base for the generated picture.

3) Code listing

The work involved using the Python programming language. The Wing-101 environment was used to develop and test the program.

The listing below shows a software implementation of the algorithm for visualizing a 20x20x20x20 tensor using pixels of varying brightness:

```
import matplotlib.pyplot as plt
import random as rd


#initilizing tensor as 400x400 matrix
a = [[rd.randint(0, 255) for i in range(400)] for i in range(400)]


#adding a bit of interface
plt.figure(figsize=(10, 10))
plt.imshow(a, vmin=0, vmax=255)
plt.colorbar(label='correlation between integer value and
    corresponding pixel brightness')
plt.title("4D tensor visualization via 2D projections")

#creating a grid for visualizing projections
plt.grid(True, color='black', linewidth=1)
plt.xticks(range(0, 401, 20))
plt.yticks(range(0, 401, 20))


#displaying the figure
plt.show()
```