Python Week 3

Huffman code

Request

1. Finish the Huffman code.

You need to write the function 1. by yourself, using the function in opencv or any image processing repository on the GitHub is NOT allowed.

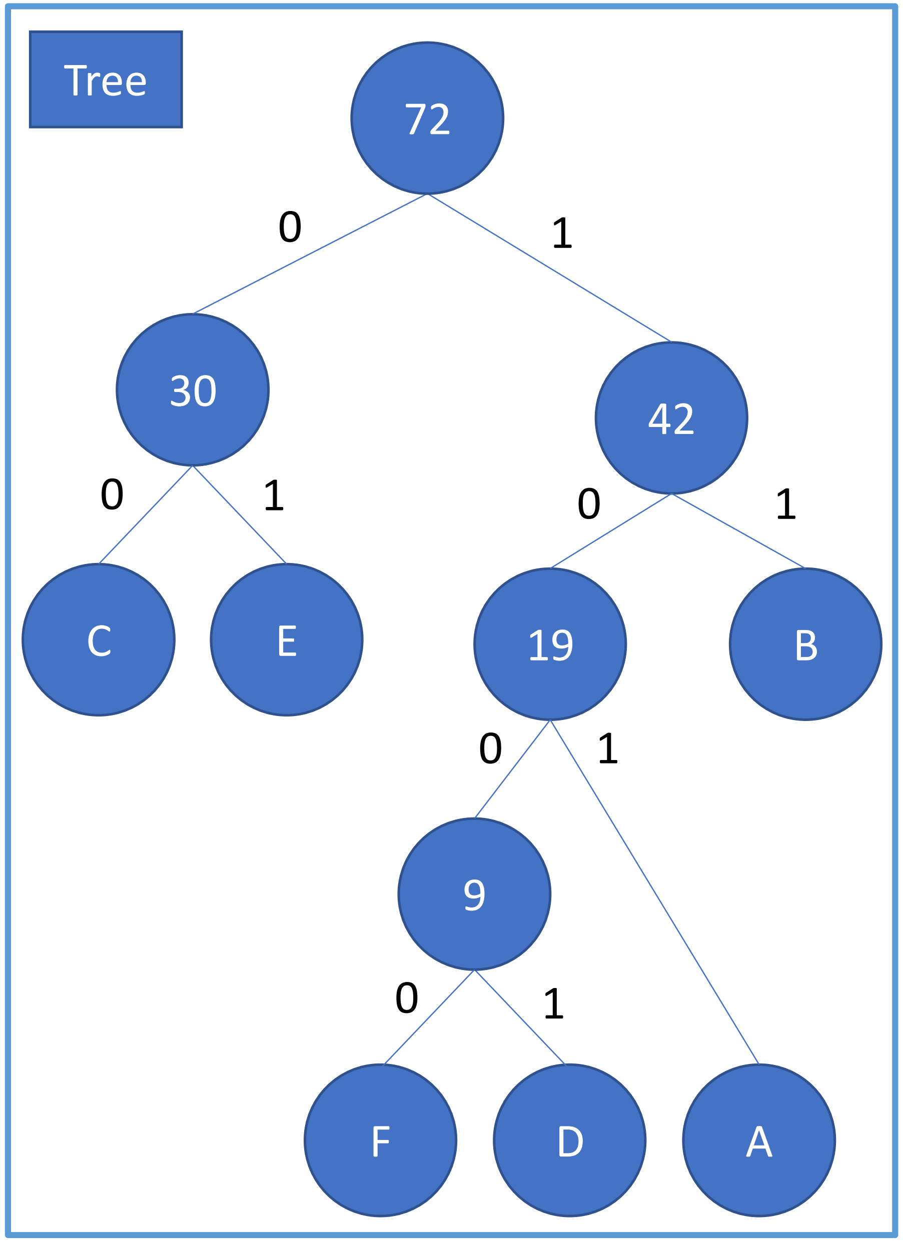
Submit your python code before 7/11.

Example

{A: 10, B: 23, C: 12, D: 6, E: 18, F: 3}

↓

A: 101
B: 11
C: 00
D: 1001
E: 01
F: 1000

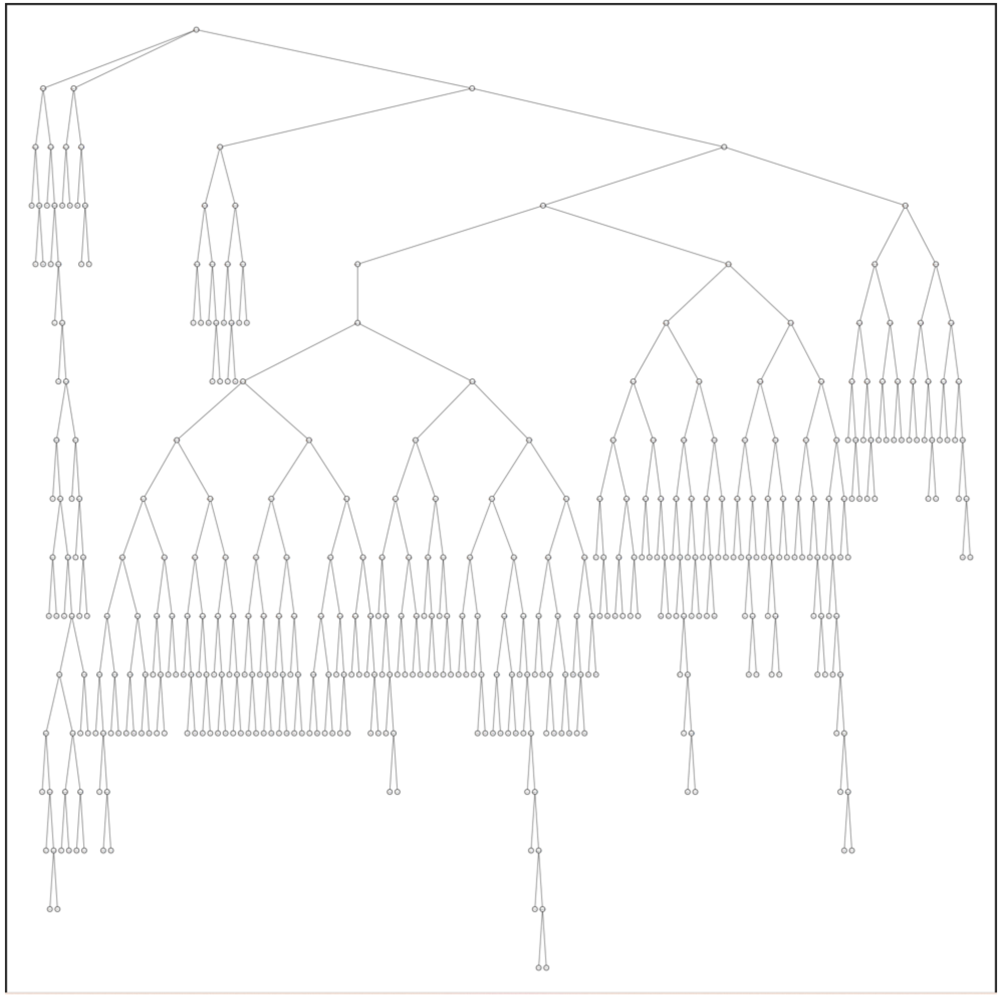


Result

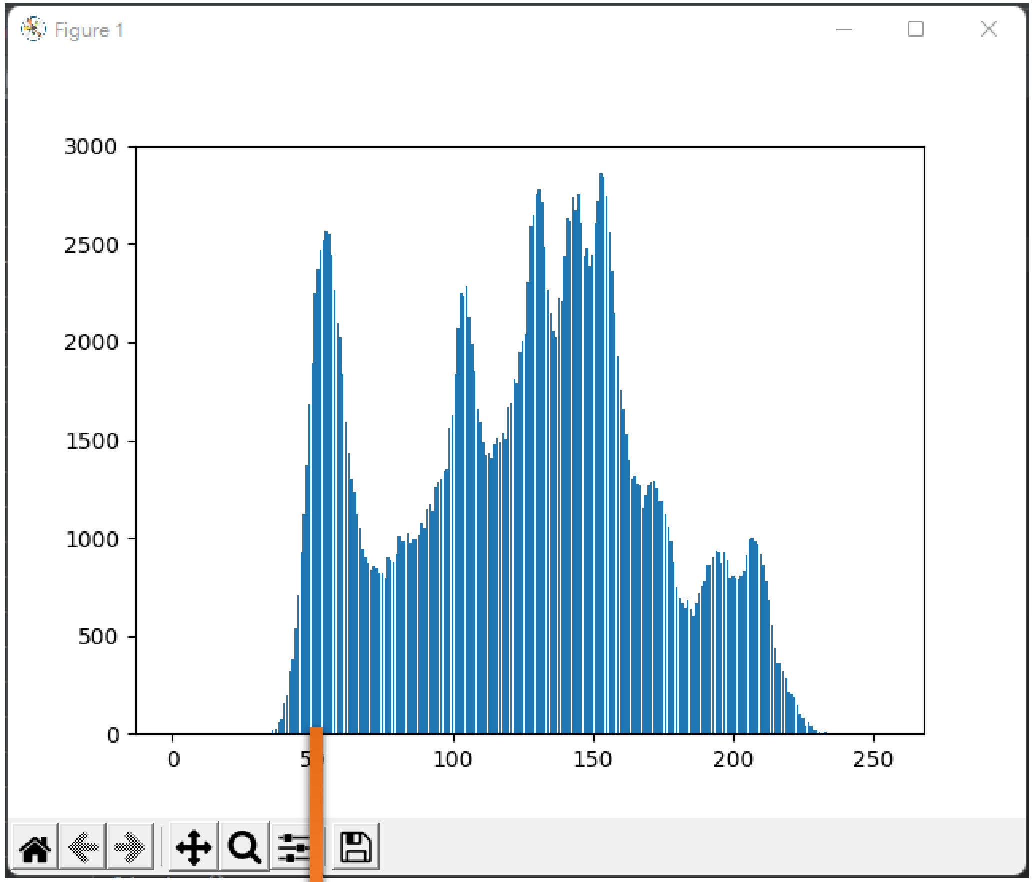
Generate the Huffman code from image.

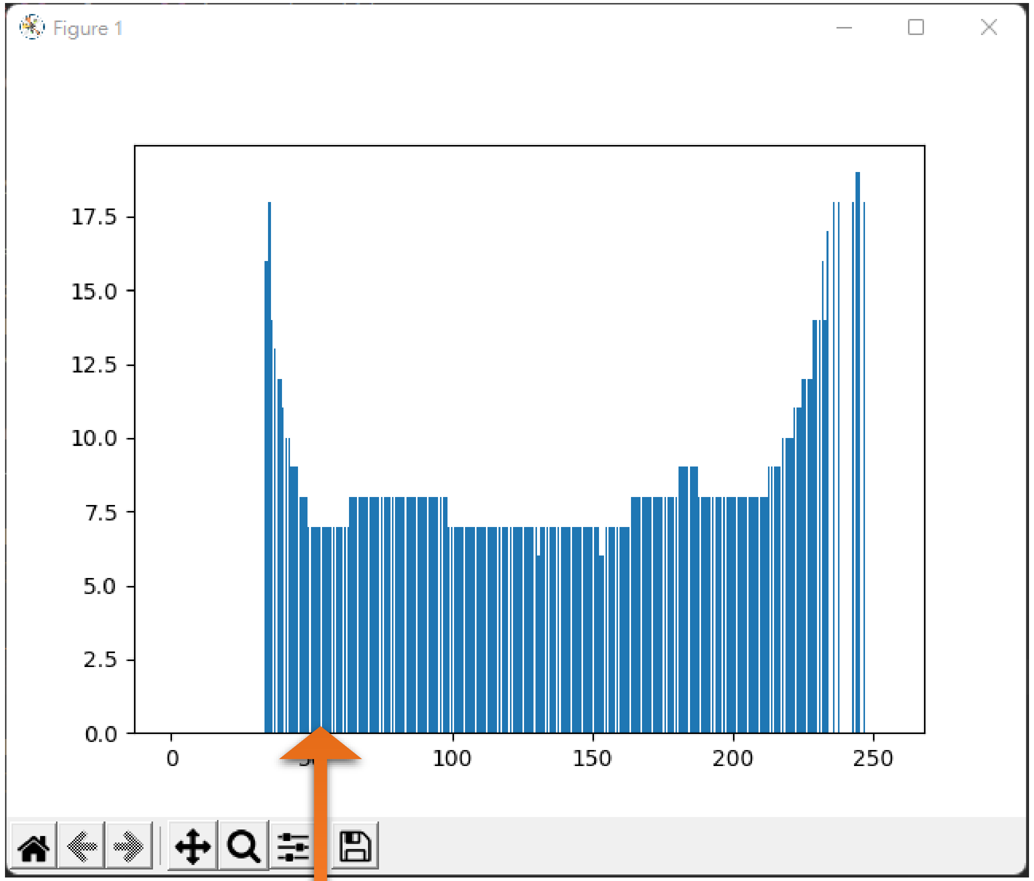
The graph on the right hand side is the Huffman tree generated from Huffman code. (igraph)

**lena.bmp and result.txt for checking your algorithm.



Result





More frequently, less bit numbers after encoding.

Reference

- https://xken831.pixnet.net/blog/post/459581308-%E9%9C%8D%E5%A4%AB%E6%9B%BC%28huffman%29%E6%A8%B9-~%E8%B3%87%E6%96%99%E5%A3%93%E7%B8%AE~

- https://www.geeksforgeeks.org/image-compression-using-huffman-coding/

- https://igraph.org/python/tutorial/latest/tutorial.html#finding-a-single-vertex-or-edge-with-some-properties

- https://towardsdatascience.com/newbies-guide-to-python-igraph-4e51689c35b4