Big Homework 2

Data Structures and Algorithms
Graphs & Trees

General mentions

• For this project you will work either in **teams of 2 people or alone.**

• The homework will be uploaded on the **Moodle** platform (fils.curs.pub.ro), for Hw2 assignment. If you encounter problems with the platform, contact by email your lab assistant. If you work in teams of 2 people, only ONE UPLOAD per team is enough.

• The homework must be submitted by **03.05.2020, at 23:59**. No late submissions will be accepted.

• You will be asked details about your solutions during the following lab.

• The final submission will contain an archive named Student1FamilyName_Student1Name_ Student2FamilyName_Student2Name_HW2 with:

· **the source files** of your project (.cpp and .h), **grouped in separate folders for each exercise** (and not contain the object files (.o), CodeBlocks project files (.cbp) or executables (.exe))

· **a README file** in which you will specify all the functional sections of the project, together with instructions for the user; additionally, if you have parts of the homework that don't work, you may offer solution ideas for a partial score on these sections.

• For all questions regarding the project, communicate by email, Microsoft teams with your lab assistant, or by using the Moodle forum

• Warning: we will use plagiarism detection software on your submissions (Stanford's tool Moss). Copied homework will be marked with **0 points.**

! Observation: The graphs and trees used will be in headers (.h) and will be generic classes (template).

1. Graphs (5p) + 1.5 p bonus

After travelling across the 7 seas, Jack Sparrow has finally found the well-known treasure of Captain Barbossa. Unfortunately, Jack must figure out how to unlock it, as the chest contains no lock, just a strange pentagon with a pentagram inside.

Tasks:

a) (1p+0.25p) The number and letters of the nodes, as well as the connections between the nodes, are written on an old parchment. Read the information from a file and construct the graph (Figure 1). If you don't know how to work with files, you can read it from the console or directly into the program.
BONUS (0.25p): Read the input data from a file.

b) (2p+0.25p) Jack thinks that a word from the pirate world should be the password, so he thinks about all possible solutions from his vocabulary. Find out which one of the following words is the first part of the code needed to unlock the chest: **pirate, rum, argh, arrgh, rug, ugh.**

Your algorithm should display: “no” if the word is found, and “yes” followed by the order of the traversal, if the word is found. Save the path in an external file or at the console.

BONUS (0.25p): Write the output data into a file.

E.g: For the word “ragu”, we display: *yes, path: 1, 0, 4, 3*

For the word “parrot”, we display *no*

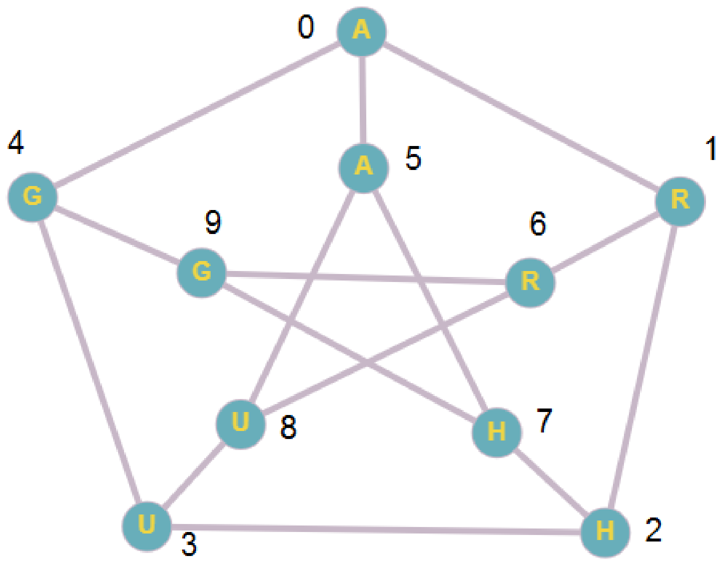


Figure 1 – chest drawing

c) It worked! Now Jack must place some precious stones on the nodes, so that we can’t find any 2 identical stones as neighbors. The available stones are sapphires (blue), rubies (red) or emeralds (green). “I can do it easily with just two colors!”

a. (1p) Good thing that Jack has the brilliant capuchin monkey with him! The monkey shows him that his argument is false – we need at least 3 types of stones to solve the problem. Write an algorithm which shows that the capuchin monkey is right.

b. (1p) Draw the graph which represents a possible solution of solving the puzzle with 3 stones.

BONUS (1p): Implement the algorithm which does the mapping between the nodes and the stones

2. Trees (5p) + 0.5p bonus

A Quad tree is used to represent crypted images. Each image can be devided into four quadrants. Each quadrant can also be divided into other four quadrants, etc. In a Quad tree, the image is represented as a parent node, while the quadrants are represented as four child nodes. If the whole image has only one colour, then the quad tree contains only one node. A quadrant has to be devided only if it has pixels of various colours. This means that the tree might not have a balanced form.

A graphical designer works with images of 32 * 32 units, this means 1024 pixels/ image. One of the operations he has to make is overlaping two images, with the purpose of obtaining a new black-and-white one. In the new image, one pixel is black if it is black in at least one of the two initial images, otherwise it is white. He wants to know how many black pixels he will have in the new image, before doing the overlaping operation, as images with too many black pixels are expensive to be printed. You have to help him by writing a program which calculates how many black pixels the final image will have. Unfortunatelly, he does not trust anybody, so he does not want to give you the initial pictures, only the Quad representations of them.

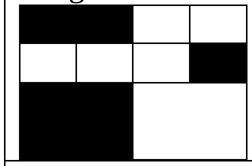
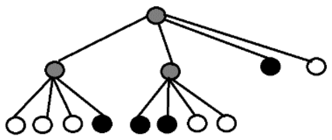
Input for your program: a pair of strings representing the two images. Thus, an image is represented by the preorder traversal of its Quad tree: letter „p" means a parent-node, letter „b" means a child-node for a black quadrant, letter „w" means a child-node for a white quadrant. The pair of strings can be read from the console or from a file.

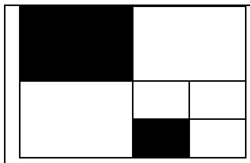
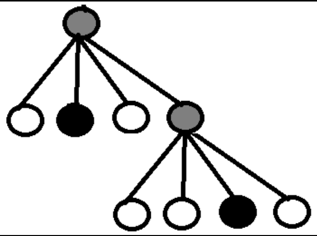
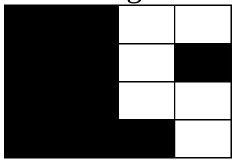
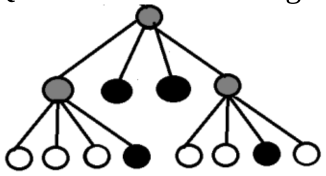
Output for your program: preorder traversal of the Quad tree of the final image and the number of black pixels contained by it, as a sum of the black pixels contained by each initial image.

Explanations:

Order of representing quadrants:

2	1
3	4

Image 1: 	Quad tree for Image 1: 	Preorder representation of image 1: ppwwwbpbbwwbw	Nr of pixels in image 1: 448
Image 2:	Quad tree for Image 2:	Preorder representation of	Nr of pixels in image 2:

		image 2: pwbwpwwbw	320
Final image: 	Quad tree for final image: 	Preorder representation of final image: ppwwwbbbpwwbw	Nr of pixels in final image: 640

Tasks:

a) (1p) Find a suitable representations for Quad trees.

b) (1p) Write a function which transforms the preorder traversal you give as an input into the representation you have chosen for Quad trees.

c) (1p) Write a function which receives as arguments two Quad trees and transforms them into another Quad tree, for the final image, based on the algorithm described in the problem.

d) (1p) Write a function which calculates the pixels for a given Quad tree representing an image.

e) (1p+0.5 bonus)Test your program with the below inputs. You should read the inputs from the console and display the expected outputs as well at the console.
BONUS (0.5p):
For working with files (reading from files and writing in files you can obtain a 0.5 bonus).

Input	Expected output
ppwwwbpbbwwbw pwbwpwwbw	ppwwwbbbpwwbw 448+320=640 (768-128 after eliminating the common black pixels)
pwwwb pwwbw	pwwbb 256+256=512
pwwwb pwwpwbwbw	pwwpwbwbb 256+128=384
ppwwwbbbpwwbw pwwpwbwbb	ppwwwbbbb 640+384=832
w pbbbw	pbbbw 0+768=768

Useful references:
fclose: http://www.cplusplus.com/reference/cstdio/fclose/
fopen: http://www.cplusplus.com/reference/cstdio/fopen/
fprintf: http://www.cplusplus.com/reference/cstdio/fprintf/
fscanf: http://www.cplusplus.com/reference/cstdio/fscanf/
http://www.cplusplus.com/doc/tutorial/files/