

Assignment 1

Deadline: 02/19/2020, 1:10pm

Unless stated otherwise, in solving those exercises you can reuse the code uploaded on the webpage of the course or portion of it. **Submit electronically via canvas both the codes and the solutions.**

Exercise 1

- Consider company *Sweets and co.* that produces and delivers croissants to coffee shops in Manhattan Upper East Side. Every morning at 5am, the truck of Sweets and co. fills his trunk with croissants in the company site, delivers to the customers, and then goes back to the company site. The location of Sweets and co. is represented by a square, while customers are represented by circles (see Figure 1, picture on the left). Formulate the problem that Sweets and co. has to solve. Clearly indicate how the weight function is computed.
- Since its business is not going well, Sweets and co. is converted into *Water inc.*, a company that installs underground pipes to deliver water. Suppose circles are again its customers, and Water inc. has to build a pipeline network to reach them. Suppose that we do not care for the robustness of the network, and the installation costs depend linearly on the length of the pipes. Also assume that pipes between any two customers can be built, following a straight line. Formulate the problem that Water inc. has to solve.
- The data from the picture is stored in the file *manhattan.txt*. The i -th row is composed of the coordinates of the i -th point, given as a pair (x_i, y_i) . The first point in the file is the company site. Implement an algorithm that gives a feasible (possibly, not optimal) solution to the problem in part a) (you may reuse portions of the code from the courseworks repository).
- Now suppose that the company switches location with one of its customers. Does the optimal solution to any of the problem above change?

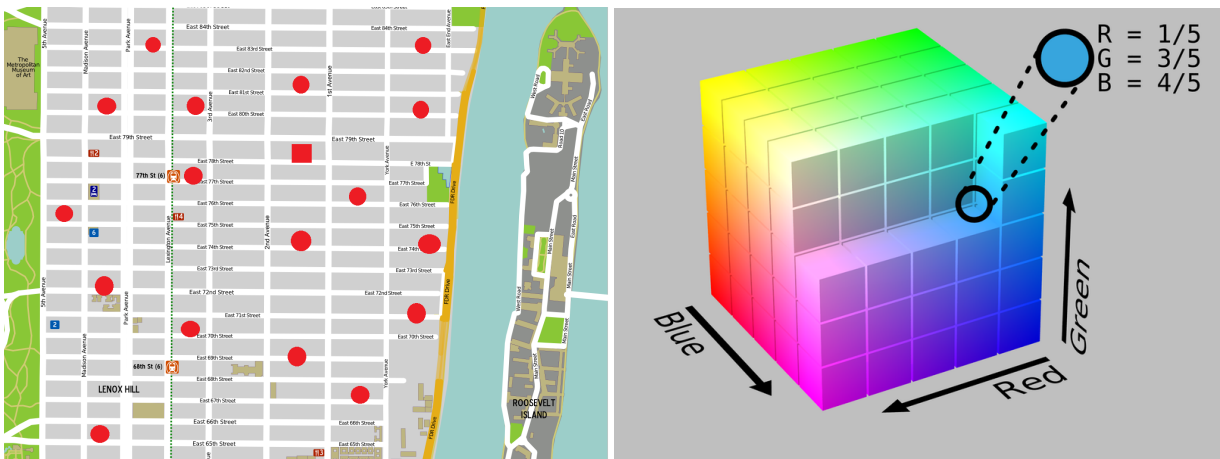


Figure 1: On the left: the map from Exercise 2. On the right: the point $(1/5, 3/5, 4/5)$ in the RGB space.

Exercise 2

Give an upper bound on the complexity of the MST-based (that is also called *Simplified Christofides*) algorithm for computing a TSP seen in class.

Exercise 3

Implement the two versions seen in class of Prim's algorithm for computing a MST. Use the code from the repository in courseworks for creating 20 instances of complete random graphs for $n = 50, 100, 200$ vertices, and report their running times on those graphs.

Exercise 4

In an image, each pixel is represented by a 3-dimensional vector in the RGB space, i.e. each component represents the amount of one of the three primary colors (red, green blue) that appears in the pixel (see Figure 1, picture on the right). Each component has value between 0 and 256. Consider the following problem. You are given an image as a set of pixels in the RGB space. Hence the amount of information you need to store is: number of pixels times 3. You now want to reduce the amount of information associated to the picture. One way of doing it is as follows: you represent the image using at most k colors, where a color is defined to be any point of the RGB space. Now it is enough to have a "catalogue" of the k colors you chose and, for each pixel, which at the k colors it corresponds to. Use the k -clustering ideas seen in class to give an algorithm that, given k and an image in input, chooses which k colors to use, and replaces each pixel with an appropriate color. (*NB: You do not have to implement the algorithm. We will see an implementation in a later class*).