



HOMEWORK 4 - CS5007

Learning objectives

- Pyplot
- Graphs
- Algorithm

The basic assignment is worth 100 points (no extra credit for this assignment). This is an individual assignment. You may discuss any aspect of this assignment with anyone, but you must type everything into an IDLE window yourself. Except where specified, you may never copy and paste from any electronic source.

Write your code and comments on a file FIRSTNAME-LASTNAME-HW4.py that you will create from scratch (there is no template). Upload it on the course website (no delay accepted).

1 Archimedean spiral (30 points)

Using Pyplot and Numpy, state a function `arc` that draws a green Archimedean spiral according to the following parameterized equation:

- $x(t) = t \cdot \cos(t)$
- $y(t) = t \cdot \sin(t)$

t should vary in $[0, 5\pi]$ (step 0.01) and the function should call `plt.plot` with arguments x and y , as well as `plt.show()` function at the end. Call the function.

2 Heart (30 points)

Using Pyplot and Numpy, state a function `heart` that draws in dashed red the following parameterized equation:

- $x(t) = 16 \sin^3(t)$
- $y(t) = 13 \cos(t) - 5 \cos(2t) - 2.5 \cos(3t) - \cos(4t)$

t should vary in $[0, 2\pi]$ (step 0.01) and the function should call `plt.plot` with arguments x and y , as well as `plt.show()` function at the end. Call the function. To see the result you should first close the window of part 1 (Archimedean spiral).

3 Graphs (30 points)

We wish to use graph theory in order to solve the following problem. A company should carry different chemical products P_1, P_2, \dots, P_k from the factory to a city. For security reasons, some products should not be carried in the same truck: $\forall i, 0 < i < k, P_i$ is not compatible with P_{i+1} . Moreover P_k is not compatible with P_1 .

1) Write a comment: how can we state an undirected graph that visually represents this problem: 1. What are the vertices, what are the edges? 2. What is the specific property of this graph?

2) Write an algorithm that, given an integer $k > 1$, returns the minimum number of trucks necessary to carry all the products.

4 Algorithm (10 points)

In ancient times, Egyptians did not use multiplication tables. Instead, they transformed any calculation into multiplication by two, and additions. The principle of Egyptian multiplication is the following: $(x * y) = 2 * (x * (y/2))$ if y is even, and $(x * y) = x + (x * (y - 1))$ if y is odd. For instance,

$$(7 * 10) = (2 * (7 * 5)) = (2 * (7 + (7 * 4))) = (2 * (7 + (2 * (7 * 2)))).$$

Write a recursive Python function that, given two strictly positive integers x and y (the arguments of the function), returns the result of $x * y$ obtained using the Egyptian method. Of course, directly multiplying x by y is allowed in your code if and only if $x \leq 2$ or $y \leq 2$.