Assignment Web Scraping

Table of contents

1 WHAT THE PROGRAM DOES

The delivery includes four files, which are web_scraping.py, plot.py, sort_lib.py and txt_files.py.

1.1 PLOT.PY FILE

The plot.py file try to load meta data from dict.txt, which is made by the web_scraping.py program. But the txt file is only made if the web_scraping.py is executes first. If dict.txt exists, the plot.py will use it for time measuring the sorting algorithms. Else if the dict.txt do not exist the plot.py makes its own test range.

Then the program uses the lists in the sorting algorithms, which is insertion sort, quick sort and bubble sort. The sorting algorithms are explained under the plot section. The program then measures the time used for each of the sorting algorithms. The time y axis is plotted for each x in range (100, 2301, 200).

The sorting algorithms are imported from sort_lib.py and the dict.txt is loaded from a function in txt_files.py.

1.2 WEB_SCRAPING.PY FILE

The program starts to check if dict.txt exist, which is the file of the images metadata from xkcd.com. That is size, name and URL of each image. If the dict.txt does not exist the program start downloading metadata from each image in “xkcd.com” and store it in the variable ‘img_liste’ as dictionary in list and then save it as dict.txt.

The program then sorts the list by using quick sort algorithm, the list is sorted descending by size. From the plot.py I found out that quick sort was the fastest sorting algorithm. The sorting algorithms are in the sort_lib.py . After the sorting the program fetches the ten largest images and print them to the screen with image name and file size.

The user then selects on of the images. If the image do not exist in the ./xkcd/ folder it will be downloaded from the xkcd.com. Else if the image exists in the folder it will be loaded from the folder. The image selected will be shown on the screen.

1.3 SORT_LIB.PY

Containing three functions, the bubble sort, insertion sort and quicksort.

1.4 TXT_FILES.PY

The file includes two functions the save_dict_to file which saves the dict.txt file including the meta data from the images and the load_dict_from_file which load the dict file.

2 LIBRARIES USED

Library	**Description**	**Use**
Requests library	Requests is standard for making HTTP requests in Python.	It is used for connecting the 'https://xkcd.com' page and to find the image size
Os library	Os library make it possible to use operating system dependent functionality, for instances read and write files.	In the program it is used to clear the screen, make directory /xkcd/, to check if a file exists and save images to directory.
Bs4 library	Beautiful Soup is a Python library for pulling data out of HTML and XML files.	It is used to find the content of the “#comic img” and to get the filename of the images.
PIL library	Pillow is the Python imaging library.	It is used in the program to open and show the images.
Random library	Random library is used as number random generator.	In the program it is used in the quick sort function the get random pivot values.
Timeit library	Timeit is for measuring the execution time for codes.	In the program it is used for time measuring the sorting algorithms.
Matplotlib library	Matplotlib.pyplot provides a graphical plotting framework.	It is used in the program to plot the time plot for the sorting algorithms.

3 THE TIME PLOT

3.1 PLOT

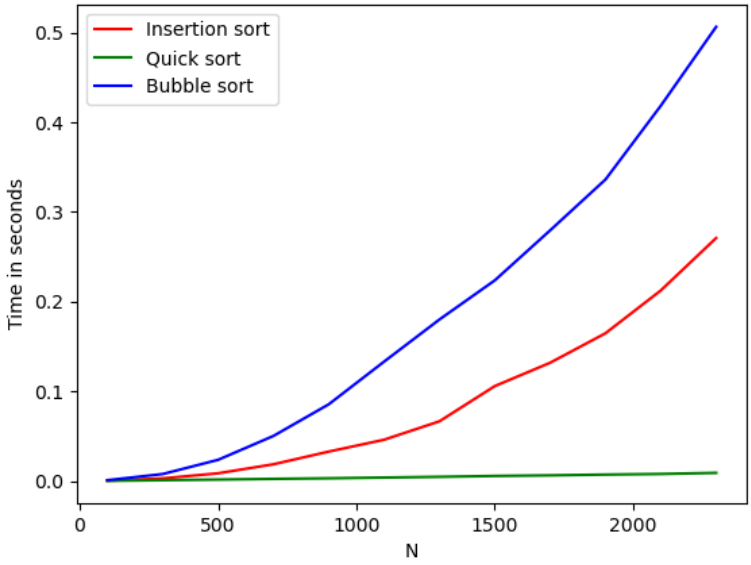

Figure 1 Y axis is the measured time and x axis is the number of sorted elements.

The plot is based on a list of 2300 elements from the image sizes and the sorting time of 100, 300, 500 and up to 2300 elements. The timeit library is used for measuring the times. In this plot it is used 1000 numbers of repetitions.

We can see on the graph that the Quick sort is the most effective sorting algorithm. In this case it seems almost linear horizontally. It should be n*log(n) graph, but it is a lot more effective than the two other methods, so it seems flat in comparison.

3.2 BUBBLE SORT

Bubble sort has two for loops and a complexity of n^2. The n is the number of items being sorted. It starts from an end and repeatedly compare each pair of adjacent values and swap them if they are not in descending order. It keeps doing this until there are no more values to swap.

3.3 INSERTION SORT

Insertion sort has a for loop and while loop and is also an n^2 sorting algorithm, but it is more effective than the bubble sort. It starts sorting from an end and then split the list in a sorted and unsorted part, the values are taken from the unsorted part and placed correctly in the sorted part.

3.4 QUICK SORT

Quick sort has an average performance of n*log(n) and is more effective than the bubble sort and the insertion sort. Quick Sort is a Divide and Conquer algorithm. It picks an element randomly as pivot and partitions the list elements into lower and higher array around the pivot number. It splits the array into smaller arrays until it ends up with an empty array, or one that has only one element. It is a recursive function and call itself until it is done.