CHU LIN
LIN.CHU@YAHOO.COM

BACKGROUND

This is a program that collects logos as it crosses Mixrank Top Ranked Websites. All sources are provided by https://mixrank.com/web/sites.

USAGE

[root]: python url_logo_crawl.py --help

-- length	length of the target list. Please use a multiple of 250.
-- worker	total number of threads generated by the main
--file	file name for the log.

FUNCTIONS AND CLASSES

<class worker>
Each worker is a thread started by the main function.
<class progess_bar>
To display the progess
<function get_url_logo>
To get the logo URL form a given URL.
<function strstr>
To find a substring in a string and return the string where the substring starts.
<function str_replace>
To replaces a portion of the string to something else
<function crawl_mixrank>
To generates a list of top ranked websites from Mixrank
<function main>:
The main function

REQUIREMENT

Python 2.7 or above

ANALYSIS AND DESIGN

The source provided by https://mixrank.com/web/sites is in HTML forms, which are human-friendly but not computer friendly. It contains useful data such as company name and URLs. It also contains needless data such as CSS and JavaScript. Therefore, the first goal is to extract and build a list of websites from the returned HTML. After analyzing the HTML, I found that all the useful data can be found in . By running a loop on <function strstr> with keyword "list-result-link", each loop the program will stop at the keyword "list-result-link". Then, <function recorder> will be called to record the company name. The length of the loop is defined by the page_limit. If page_limit equals 50, the returned HTML is assumed to have 50 useful data.

After having a list of websites, the program will start to collect logos that associate with the websites. By definition, Logo is defined to be a symbol that is adopted by an organization to identify itself. Each website contains a lot of images. It

might require a huge computation power to determine which image is the symbol of the organization. However, after some brainstorming, I found that the question might not be that complicated. There is a dedicated location for all websites to place its logo. Then, the question reduces down to extracting the shortcut icon of the website. By using <function strstr> again with keyword "shortcut icon" or "image/x-icon", the program now has the URL to the websites logo. Some web servers store their favicon on the same origin but some don't. Thus, in the case that the server stores its favicon in the same origin, the program adds the root URL automatically. In the case that the website has neither shortcut icon nor image/x-icon, the crawler will use the default location which is https://example.com/favicon.ico.

SCALABILITY

At this point, the goal of collecting logos has achieved. The last question is the performance or the speed of the execution when processing a lot of websites. In fact, most of the time is wasted while making the request and waiting for the reply. Thus, by having multiple threads (workers), when one thread is waiting for the reply, other threads are still able to fire new requests.

PERFORMANCE

CPU: Intel Core i5
Mem: 8GB 1600MHz DDR3
OS: OS x v10.11

	Worker = 1	Worker = 5	Worker = 10	Worker = 100
Length = 10	14.73 s	-	-	-
Length = 100	-	54.47s	-	-
Length = 1000	-	-	490s	-
Length = 10000	-	-	-	>1800s

The relation between length and worker is not linear because of the limitation of the hardware. The testing system uses an Intel Core i5 duo-core. With hyperthreading enabled, it is able to simulate 4-thread simultaneously. Thus, anything larger 4 would have no physical impact but to reduce the request time (see scalability for explanation).

CAUTION

Please don't generate 100 workers for 1 URL. This is meaningless.