**Web Crawler Project**

We would like for you to complete a project. There is a two week deadline on this, and we ask that you let us know how much time you spent working on this when you submit your solution.

Feel free to reach out to Ryan if you have any questions.

Please write a very simple web crawler where you fetch URLs in parallel. You will take a starting URL as an input and then print to stdout: each crawled URL and an indented list of associated links on that page, for example:

$ ./crawler.py [http://www.rescale.com](http://www.rescale.com/)

[http://www.rescale.com](http://www.rescale.com/)

<http://www.rescale.com/booking/>

<http://blog.rescale.com/events/>

<http://insidehpc.com/2017/01/video-rescale-night-showcases-hpc-cloud/>

<http://www.digitaleng.news/de/rescale-hosts-first-ever-rescale-night-san-francisco/>

<https://blog.rescale.com/rescale-night-recap/>

<http://www.rescale.com/about/>

<http://www.rescale.com/investors/>

[http://blog.rescale.com](http://blog.rescale.com/)

<http://www.rescale.com/jobs/>

<http://www.rescale.com/legal/>

<http://www.rescale.com/booking/>

<http://blog.rescale.com/events/>

<http://insidehpc.com/2017/01/video-rescale-night-showcases-hpc-cloud/>

<http://www.digitaleng.news/de/rescale-hosts-first-ever-rescale-night-san-francisco/>

<https://blog.rescale.com/rescale-night-recap/>

<http://www.rescale.com/about/>

<http://www.rescale.com/investors/>

[http://blog.rescale.com](http://blog.rescale.com/)

<http://www.rescale.com/jobs/>

<http://www.rescale.com/legal/>

<http://blog.rescale.com/events/>

[http://www.rescale.com](http://www.rescale.com/)

<http://blog.rescale.com/>

<https://blog.rescale.com/archives/>

<http://blog.rescale.com/events/>

<https://blog.rescale.com/events/category/webinars/>

<https://blog.rescale.com/reup/>

<http://blog.rescale.com/events/>

<http://blog.rescale.com/ja/events/>

...

So in the above example, on the [http://www.rescale.com](http://www.rescale.com/) page, there are the following links:

<http://www.rescale.com/booking/>

<http://blog.rescale.com/events/>

<http://insidehpc.com/2017/01/video-rescale-night-showcases-hpc-cloud/>

<http://www.digitaleng.news/de/rescale-hosts-first-ever-rescale-night-san-francisco/>

<https://blog.rescale.com/rescale-night-recap/>

<http://www.rescale.com/about/>

<http://www.rescale.com/investors/>

[http://blog.rescale.com](http://blog.rescale.com/)

<http://www.rescale.com/jobs/>

<http://www.rescale.com/legal/>

Note you do not need to worry about nesting at all, each URL visited is printed without indent and the links in the page are printed with one indentation (spaces or tab).

A few guidelines:

- Only take links from <a href> tags and do not worry about handling all the different link formats (absolute, relative, anchor, etc.). To make things easier, you can just handle absolute URLs (starting with "http" and "https") if you want and ignore other formats.

- We would prefer a solution in either Java or Python but you are welcome to use whatever language you want. A couple of tips if you decide to use Python:

    - I recommend managing the URL-fetching threads with concurrent.futures.ThreadPoolExecutor

    - I recommend fetching URLs with the python requests package (unless you have another method you are already familiar with).

- And the most important part: Code readability and cleanliness is more important than cleverness.