ADVANCED PYTHON HOMEWORK 6

OK Google

Choose one of the following tasks and solve it. Every task is worth 5 points.

Lab teacher's notes: Do not forget to test your solutions with a reasonable amounts of tests, and include them in the homework. At least a few tests are required for any correct homework. Time measurements and determining how large inputs are still feasible for your solution is also strongly recommended. If you ever make assumptions that are not specified in the exercise, make sure that they are well documented.

Exercise 1. Implement a function `crawl(start_page, distance, action)` which reads a page at the address `start_page`, calls the function `action` whose argument is the page content and then calls the function for other pages whose links can be found on this page.

As there can be many links, search depth is limited by the parameter `depth`. Also make sure not to process the same page twice. Implement the solution as an iterator, which returns tuples `(url, result_of_the_function_action)`.

You are free to test/implement several variants of `action`, but we ask you to at least implement an action that outputs all sentences containing the word *Python. Lab teacher's comment: Please note that some sentence parsing and extraction is expected in your solution.*

Exercise 2. Implement a website monitoring system that checks if a page has changed its content. We assume that our program can monitor more than one page; we also assume that checking takes place periodically (e.g. every 1 minute).

In this task we assume that page layout changes rarely, only individual elements of this layout are changed, so if the program detects a change, it must return only what has changed.

Exercise 3. Write your own web indexing system which:

- browses the pages and stores the number of occurrences of particular words on each page;
- behaves like a Python dictionary, where the key is a word and a value is a list of pages on which this word occurs (or an empty list). Pages should be sorted in a descending order by the given number of occurrences.

  You can also propose and implement your own page ranking strategy.

We assume that the program starts indexing by the page (or a list of pages) that is indicated as a parameter of our function or method, and then also indexes all pages that can be reached from the starting pages via links and href in no more then a predetermined maximum number of steps.