Стажеры 2023 - тестовое

# 1) задача на алгоритмы

*(язык - любой на выбор. Предпочтительнее - javascript / typescript)*

**дано** - массив путей к папкам 1-го уровня (задано в константе)

В папках находятся либо вложенные папки, либо файлы с разными расширениями.

Не бывает, чтоб в папке были одновременно и файлы и вложенные папки (то есть файлы всегда лежат на самом нижнем уровне)

**надо**

* уровень 1 (на троечку) - сформировать массив всех путей к конечным папкам, содержащим файлы с расширением .js (если в папке есть файлы и других расширений кроме .js, такую папку берем. если в папке только файлы не .js , такие папки не берем)
* уровень 2 (на четверку) - кроме пути к папке, также хранить кол-во файлов с расширением .js в этой папке
* уровень 3 (на отлично) - для заданного (в константе) числа N, полученный на шаге (1) массив подпапок разделить на N массивов, так чтобы все полученные массивы содержали более или менее одинаковое кол-во элементов.

(не обязательно находить самый оптимальный вариант с максимально равным разделением. Можно ограничиться более простым и быстрым алгоритмом, выдающим более-менее приближенные к оптимальному варианты)

*пример:*

|  |  |  |
| --- | --- | --- |
| структура папок на диске  A  A1-1  A2-1  file1.txt  file2.js  A1-2  file3.js  file4.js  file5.js  A1-3  readme.txt  B  B1-2  file11.js  file12.js  file13.js  B1-3  file15.js  B1-4  file8.js  readme.txt | вход:  [ A, B ]  *(массив с перечислением путей к корневым папкам)*  N = 2 | выход (задачи 1 и 2):   * A \ A1-1 \ A2-1 (1) * A \ A1-2 (3) * B \ B1-2 (3) * B \ B1-3 (1) * B \ B1-4 (1)   выход (задачи 3):  [1]   * A \ A1-1 \ A2-1 (1) * B \ B1-2 (3)   [2]   * A \ A1-2 (3) * B \ B1-3 (1) * B \ B1-4 (1) |

# 2) опционально. кто сделает - молодец

написать простой автотест:

- задача - на Вилдберис найти и вывести первые 10 штук транспортиров (такая круглая линейка для измерения углов), по возрастанию цены.

- подробно:

* открыть страницу Вилдберис.
* в строке поиска ввести слово **транспортир** и нажать Ентер.
* на странице с результатами нажать на выпадающий список сортировки и выбрать сортировку по цене по возрастанию.
* найти на странице и вывести первые 10 элементов - название и цена.

как и на чем писать:

а) скачать докер образ с хромом и puppeteer,

развернуть,

по аналогии с примером написать свой тест.

для тех, кто в первый раз:

* скачать docker desktop для windows.
* поставить wsl2 - докер при установке выведет ссылку, откуда скачать.
* скачать образ окружения для автотестов - **zenika/alpine-chrome:with-puppeteer**
  + сайт с образом –
  + <https://hub.docker.com/layers/zenika/alpine-chrome/with-puppeteer/images/sha256-855f9ac9c808ac0a567efe4c0a19f93442cb58ad3bdf17533e958003a2846e0b>
  + команда для скачивания - docker pull zenika/alpine-chrome:with-puppeteer
* выбрать (создать) каталог на локальном диске, где будете писать тест.
* внутри него создать каталог src
* в него списать и положить файл -   
  <https://github.com/jlandure/alpine-chrome/blob/master/with-puppeteer/test/pdf.js>
* попробовать запустить пример командой:  
  docker container run -it --rm -v (полный путь к папке)\src:/usr/src/app/src --cap-add=SYS\_ADMIN zenika/alpine-chrome:with-puppeteer node src/pdf.js
* в результате в папке src должен создасться файл .pdf

б) любой другой вариант, на чем умеете, лишь бы был код теста, который делает то, что надо, и краткий мануал как это запустить.

—

*если будут вопросы на понимание задания (что требуется сделать) - можете уточнять все вопросы через Анну Шаталину.*