

Задание

Требования:

Python 3



Клиент должен корректно отображаться в браузере Mozilla Firefox 40+

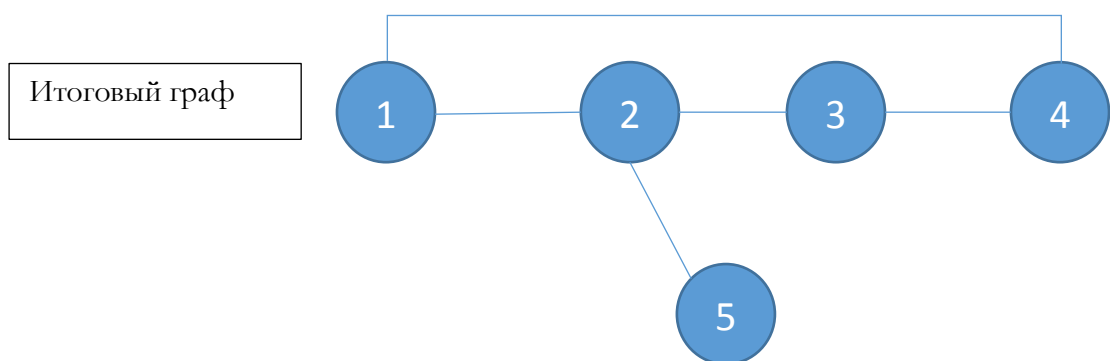
Условие:



Исходными данными для построения графа является Excel-файл. Структура Excel-файла представлена в примере. На графе должны в виде кружков с номером отображаться «узлы» из файла, в виде линий «ветки», их соединяющие. Между двумя узлами может быть несколько веток.

Пример:

| Исходные данные Excel-файла | | |
|-----------------------------|--------|------------|
| Ветки | | Узлы |
| Узел 1 | Узел 2 | Номер узла |
| 1 | 2 | 1 |
| 2 | 3 | 2 |
| 3 | 4 | 3 |
| 4 | 1 | 4 |
| 2 | 5 | 5 |



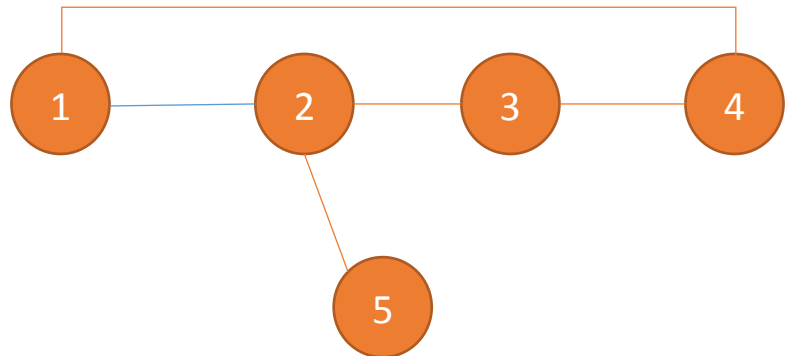
Необходимо написать небольшое серверное web-приложение (HTTP) с тонким клиентом. Данное приложение возвращает единственную HTML-страницу по корневому ресурсу на 2016 порту, которая содержит в себе следующий функционал:

- Есть возможность загрузки Excel-файла (по нажатию на кнопку или через Drag&Drop область).
- На странице отображается «список загрузок» по времени, есть возможность выбора элемента.
- На странице есть область, в которой отображается граф, сформированный из данных Excel-файла.
- При выборе элемента из «списка загрузок» отображается граф, соответствующий данным элемента.

Сервер должен сохранять результаты загрузок (после перезагрузки сервера все загруженные результаты должны быть доступны).

Дополнительная задача: подсветка в клиенте любым цветом наиболее длинного пути графа из неповторяющихся элементов (весы веток графа одинаковые). Концом и началом пути всегда является узел. При совпадении длин нескольких путей выделить любой из максимальных.

Пример выделения максимального пути из неповторяющихся элементов
(путь отмечен оранжевым)



Примечание:

- разрешается использование сторонних открытых библиотек;
- код должен быть выложен на GitHub;
- граф должен быть масштабируемым (необходимо иметь возможность прочитать номер любого узла и граф целиком);
- если есть повторяющиеся номера узлов в Excel-файле, то игнорировать их;
- если в ветке указан номер несуществующего узла, игнорировать данную ветку;
- в ReadMe должна быть описана последовательность запуска сервера и системные требования, если такие имеются.

Что оценивается:

- работоспособность кода;
- читаемость python-кода, соответствие PEP-8;
- выбор библиотек;
- декомпозиция.