Затраченное время – три дня.

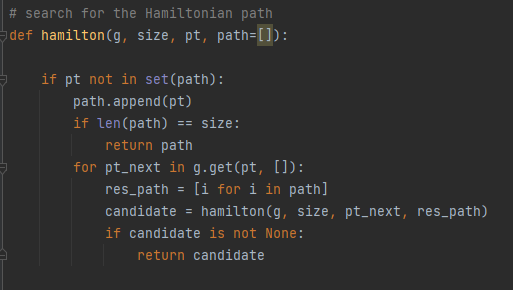
Условие:

[Дано расписание переездов](https://drive.google.com/file/d/0BwuqCIWvjy0_Mzd3UXI4T1RJTVc4aWZSS3FHU29wWWFIa2ZV/view?usp=sharing&resourcekey=0-D0tExCdVgaa1qYFKDht2pQ) поездов между несколькими станциями в формате “номер поезда; станция отправления; станция прибытия; стоимость; время отправления; время прибытия” (гарантированно, что нет переездов дольше суток) . Необходимо получить “лучшие” варианты (несколько, если это возможно) проезда между всеми станциями так чтобы побывать на каждой станции 1 раз. Запросы на лучшие варианты:

* Лучшие по цене
* Лучшие по времени

Варианты проезда между всеми станциями будем искать с помощью Гамильтонова пути. Гамильтонов путь (или гамильтонова цепь) — путь (цепь), содержащий каждую вершину графа ровно один раз.

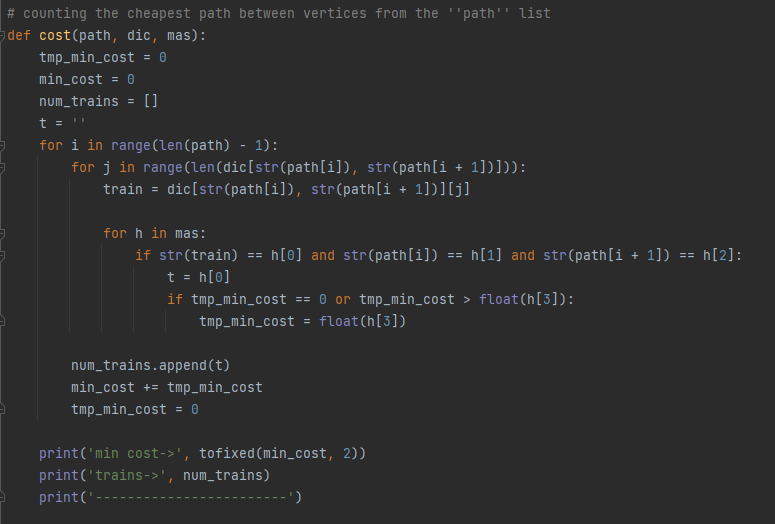
Функция hamilton получает матрицу смежности(''g''), заданной как словарь, количество вершин(''size''), начальную вершину(''pt'') и пустой массив. Возвращает функция гамильтонов путь.



Функция **cost** вернет стоимость самого дешевого пути между всеми станциями. На вход функция получает Гамильтонов путь(''path''), матрицу смежности в виде словаря(''dic'') и массив данных(''mas''), считаного с исходного файла.

Между двумя станциями выбираем поезд, у которого стоимость поездки самая дешевая и это значение добавляем в итоговую переменную ''min\_cost''.

После обхода всех станций на консоль будет выведено стоимость пути и поезда, на которых нужно перемещаться.



Функция **time** вернет стоимость самого быстрого пути между всеми станциями. На вход функция получает Гамильтонов путь(''path''), матрицу смежности в виде словаря(''dic'') и массив данных(''mas''), считаного с исходного файла.

На первой переправе между станциями в итоговую переменную ''min\_time'' мы кладем значение самого быстрого поезда из всех предложенных, потому что мы на этом шаге не учитываем время ожидания поезда. На остальных итерациях учитываем время ожидание следующего поезда- выбираем поезд, у которого сумма ожидания и поездки будет наименьшей среди всех.

Итоговый путь и время в пути выводим на консоль.

