\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* Python+IDLE \*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi>

Sublime Text 3 <https://download.sublimetext.com/Sublime%20Text%20Build%203126%20x64%20Setup.exe>

Материалы, задания и комментарии будут выкладываться в репозитарий GitHUB:

<https://github.com/dremdem/maykor_python_learn>

ДЗ:

<https://github.com/dremdem/maykor_python_learn/blob/master/Lesson6/lesson6_timing.txt>

<http://yaroslavl.taxi-mezhgorod.ru/#advantages> Ярославль

|  |
| --- |
| Есть два списка чисел, сумма элементов которых и длина одинаковы (например 2 списка из 10 элементов сумма в которых 100). |
|  |

|  |
| --- |
| С помощью «передвижения» элементов нужно изменить второй список т.о., чтоб сумма модулей поэлементной разности между списками была минимальной. Элементы второго списка можно двигать в любую сторону на любое число шагов, можно помещать в одну ячейку более чем один исходный элемент, но изначальная очередность элементов должна сохранятся. |
|  |

|  |
| --- |
| Пример 1. |
|  |

|  |
| --- |
| Список 1 |
|  |

|  |
| --- |
| 0, 0, 10, 70, 20, 0, 0, 0, 0, 0 |
|  |

|  |
| --- |
| Список 2 |
|  |

|  |
| --- |
| 0, 0, 0, 0, 0, 0, 20, 70, 10, 0 |
|  |

|  |
| --- |
| Сумма модулей поэлементной разности 200 |
|  |

|  |
| --- |
| Решением будет сдвиг списка 2 влево |
|  |

|  |
| --- |
| 0, 0, 20, 70, 10, 0, 0, 0, 0, 0 |
|  |

|  |
| --- |
| Сумма модулей поэлементной разности 20 (менять 10 и 20 местами нельзя , т.к. последовательность должно сохранится) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Пример 2 |
|  |

|  |
| --- |
| Список 1 |
|  |

|  |
| --- |
| 10, 0, 0, 10, 40, 20, 0, 10, 0, 10 |
|  |

|  |
| --- |
| Список 2 |
|  |

|  |
| --- |
| 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 |
|  |

|  |
| --- |
| Здесь решением будет список аналогичный первому и разность списков будет равна нулю. |
|  |

|  |
| --- |
| Если расписать по элементам, то в результате сдвигов элементы списка 2 расположатся так |
|  |

|  |
| --- |
| 10, 0, 0, 10, 10 10 10 10, 10 10, 0 , 10, 0, 10 |
|  |

|  |
| --- |
| Визуально процесс перестановки можно представить как перестановку «перегородок» в целевой последовательности. Т.е. изначально перегородки между каждым элементом, но их можно переставить так чтоб в любой ячейке было или несколько элементов или вообще не было (в этом случае считается что там нуль). |
|  |

|  |
| --- |
| В общем случае может быть несколько лучших решений, в этом случае берется любое. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| P.S. |
|  |

|  |
| --- |
| Перебирать все возможные перестановки, для каждой вычисляя разность, решением не считается. Для 10 элементов это еще прокатит, а для 100 уже умрет, так как там количество перестановок несильно меньше факториала получается. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Приз за лучшее решение коньяк )) |
|  |

GIT



