**Звіт**

У програмі реалізований пошук найкоротшого шляху через алгоритм Левіта.

Алгоритм:

1)Створити масив (у нашому випадку список)для збереження шляхів до точок;

2) На початку ініціалізувати його зі значеннями infinity(у програмі таке значення позначає None), а відстань до початкової точки позначити 0.

3)Створити масив(список) ‘предків’ точок у найкоротших шляхах до них, з початковими значеннями відповідними точками для кожної позиції.

4)Створити множини для точок до яких обраховується відстань(нехай тут м1(у оригінальному алгоритмі м1 – черга, у програмі список)), до яких ще не була обрахована відстань(нехай тут м2) і для яких уже виконані обрахунки(нехай тут м0)

(У програмі є ще 2 кроки, конвертації заданого ландшафту зі списку у numpy.array і отримання його розмірностей. Також створюється допоміжна множина для ребер, довжина яких обрахована.)

5)Ітерувати через множину м1, для кожного елементу, переміщувати його в м0, знаходити суміжні точки, ітерувати через них і виконувати наступні дії:

А)Обраховувати відстань ребра з цією суміжною точкою;

Б)Якщо ця суміжна точка у множині м2,

то присвоїти значення знайденої відстані, якщо відстань до точки з м1 None або значення суми відстані до цієї точки і довжини ребра між даними точками. Видалити суміжну точку з множини м2 і перемістити у м1. Повторити для всіх суміжних точок.

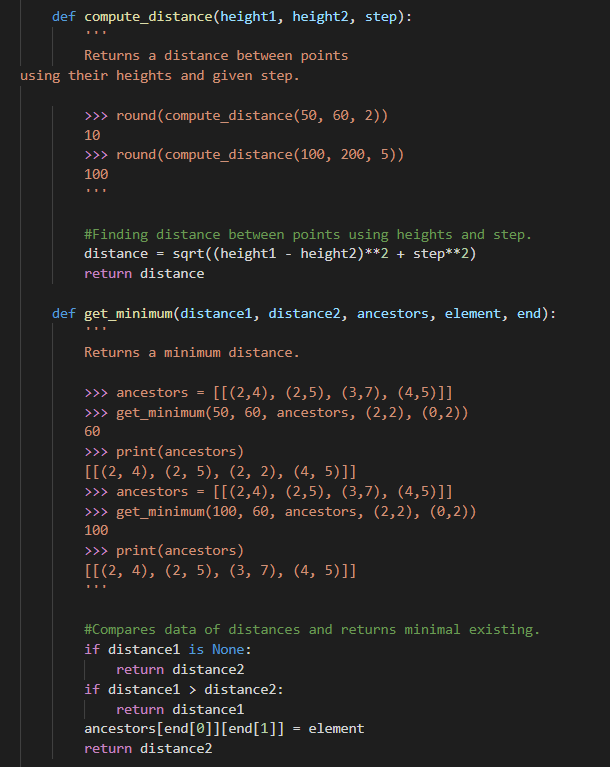
В)Якщо точка в м1, оновити її значення мінімальним значенням серед старої обрахованої відстані і нової.

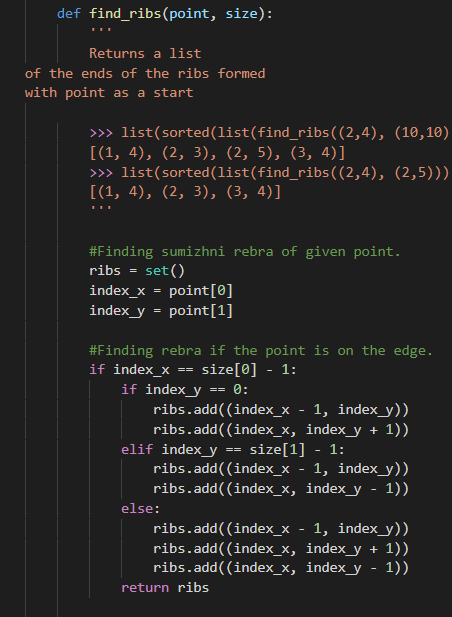
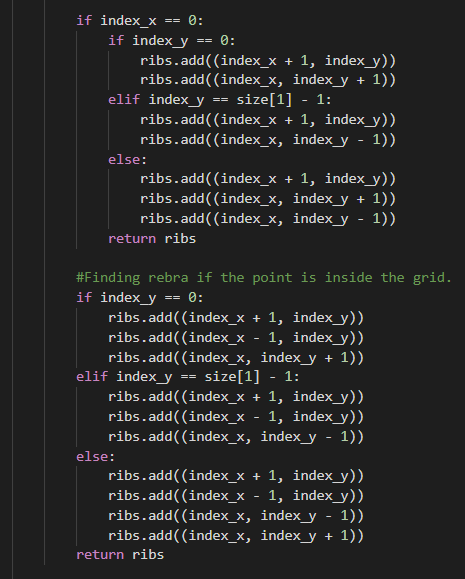
Г)Якщо точка у м0, оновити її значення мінімальним значенням серед старої обрахованої відстані і нової.

(Якщо значення предка для точки з м1, яку ми розглядаємо, співпадає зі значенням суміжної точки, ребро з якою ми обраховуємо, існує варіант розвитку подій при якому пройти з такої суміжної точки туди назад до точки з м1 швидше ніж зі ‘старого предка’. Тоді програма шлях від предка зациклюється і це унеможливлює пошук необхідного шляху. Для цього у програмі передбачена додаткова умова.)

6)Результат конвертується у одинарні індекси і повертається як список.

У програмі додатково реалізовані функції для визначення мінімальної відстані, пошуку суміжних точок і визначення відстані:



Виконали:

Броницький Михайло  
Роман Кипибіда  
Трескот Вадим  
Ростик Сидор  
Гоєв Олексій