# ЗМІСТ

|  |  |
| --- | --- |
| **ВСТУП** | 2 |
| [**РОЗДІЛ 1.**](#_Toc42807121)Опис бази практики | 4 |
| [**РОЗДІЛ 2.**](#_Toc42807121)Постановка задачі дипломного проектування та її формалізація | 6 |
| **РОЗДІЛ 3.** Вибір та обґрунтування методів і засобів розв’язання завдання | 8 |
| 3.1 Python | 8 |
| 3.2 Numpy | 10 |
| 3.3 Matplotlib | 11 |
|  |  |
| **РОЗДІЛ 4.** Опис реалізації завдання | 12 |
| **ВИСНОВКИ** | 15 |
| **ПЕРЕЛІК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ** | 16 |

# ВСТУП

Поняття алгоритму інтуїтивно зрозуміла та часто використовується в математиці та комп’ютерних науках. Говорячи неформально, алгоритм – це довільна коректно визначена обчислювальна процедура, на вхід якої подається деяка величина або набір величин, а результатом виконання якої є вихідна величина або набір значень. Таким чином, алгоритм є послідовністю обчислювальних кроків, які перетворюють вхідні величини у вихідні. Алгоритм можна також розглядати як інструмент, який призначений для вирішення коректно поставленої обчислювальної задачі. У постановці задачі в загальних рисах визначаються відношення між входом та виходом. В алгоритмі описується конкретна обчислювальна процедура, за допомого якої можна досягнути виконання вказаних відношень.

В комп’ютерних науках сортування є основною операцією (у багатьох програмах вона використовується в якості проміжного кроку), в результаті чого з’явилось багато якісних алгоритмів сортування. Вибір найбільш адекватного алгоритму залежить від багатьох факторів, в тому числі й від кількості елементів для сортування, від їх порядку у вхідній послідовності, від можливих обмежень, які накладаються на членів послідовності. Кажуть, що алгоритм є коректним, якщо для кожного входу результатом його роботи є коректний вивід. Тоді коректний алгоритм розв’язує дану обчислювальну задачу. Якщо алгоритм некоректний, то для деяких входів він може взагалі не завершити свою роботу або видати відповідь, яка відрізняється від очікуваної.

Алгоритми є життєво необхідними складовими для рішення будь-яких задач з різноманітних напрямків комп’ютерних наук. Алгоритми відіграють ключову роль у сучасному розвитку технологій. Тут можна згадати такі розповсюджені задачі, як:

* Розв’язання математичних рівнянь різної складності, знаходження добутку матриць, обернених матриць;
* Знаходження оптимальних шляхів транспортування товарів та людей;
* Знаходження оптимальних варіантів розподілення ресурсів між різними вузлами (виробниками, верстатами, працівниками, процесорами тощо);
* Знаходження в геномі послідовностей, які співпадають;
* Пошук інформації в глобальній мережі Інтернет;
* Прийняття фінансових рішень в електронній комерції;
* Обробка та аналіз аудіо та відео інформації.

Метою практики є освоєння нових методів дослідження, та розробки програмного забезпечення. Огляд традиційних технологій, вивчення нових технологій для розробки ефективного алгоритму для розв’язання задачі будівництва залізної дороги на новому континенті.

Проведення практики включало: обрання теми проекту для практики, вивчення отриманого матеріалу, виконання і перевірка індивідуального проекту а також на завершенні практики, презентація виконаного проекту з відповідною темою.

Для практики я обрав тему Railroad Tycoon

Предметом дослідження є розробка програмного продукту для підприємця тридцяти років, який збирається розпочати будівництво залізної дороги на новому континенті згідно всіх його вимог у вказаний термін за допомогою технологій Python та додаткових бібліотек

# РОЗДІЛ 1

## ОПИС БАЗИ ПРАКТИКИ

Проходження практики відбувалося в IT-компанії “ELEKS”. Елекс - українська IT-компанія, що надає повний спектр програмної інженерії, консультаційних послуг та забезпечення контролю якості. Компанія була створена у 1991 році як мале комунальне підприємство і з того часу розрослася до однієї з найбільших ІТ-компаній України. На початку свого існування компанія займалася розробкою власних програмних продуктів для електроенергетичних підприємств. З 1999 року "ELEKS Software" почав працювати на ринку аутсорсингу, а сьогодні послугами компанії користується більше 200 компаній-замовників у всьому світі. Компанія бачить свою місію у тому, щоби допомагати лідерам ринку та компаніям, які претендують на лідерство в галузі технологій, перетворювати креативні ідеї в реальну економічну перевагу; створювати унікальні та надійні рішення для забезпечення безпрецедентного росту бізнесу. Цінності ELEKS – досконалість, відповідальність, професійна команда, інновації і партнерство.

*Інформація про IT-компанію:*

|  |  |
| --- | --- |
| Назва: | ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ ЕЛЕКС |
| Дата реєстрації: | 12.03.1998 |
| Форма власності: | Недержавна власність |
| Контактна інформація: | *Адреса:* Україна, 79060, Львівська обл., місто Львів, ВУЛИЦЯ НАУКОВА, будинок 7, корпус Г |
| *Телефон:* 22971251 |
| *Факс:* 22447002 |
| Керівник: | Коновалов Петро Олександрович |
| Види діяльності: | *Основна:*  62.01 Комп'ютерне програмування |
| *Інші:*  58.29 Видання іншого програмного забезпечення  62.02 Консультування з питань інформатизації  62.09 Інша діяльність у сфері інформаційних технологій і комп'ютерних систем  74.30 Надання послуг перекладу |

Компанія ELEKS одна із ведучих компаній з питань аутсорсингу, яка співпрацює з малим та середнім бізнесом та корпоративними клієнтами, щоб підвищити їх цінність за допомогою експертної інженерії програмного забезпечення, дизайну продуктів, забезпечення якості та консультаційних послуг. Компанія допомагає різним галузям, включаючи логістику, роздрібну торгівлю, сільське господарство, охорону здоров’я та уряд, цифровими інноваційними технологіями.

# РОЗДІЛ 2

## ПОСТАНОВКА ЗАДАЧІ ДИПЛОМНОГО ПРОЕКТУВАННЯ ТА ЇЇ ФОРМАЛІЗАЦІЯ

Комп’ютерні алгоритми глибоко проникли в усі сфери нашого життя. Вони створюють дивовижні музичні симфонії, аналізують юридичні документи, пишуть новини та керують автомобілями. Їхнє поширення одночасно лякає та... вселяє надію. Чи може комп’ютер передбачати потенційний музичний хіт? Чи замінять алгоритми поетів і режисерів? Чи зможуть узяти під контроль усю медичну галузь? І яку роль спроможні відігравати в політиці?

Сьогодні інтернет-простір вирішує багато проблем в інформаційному плані. Важливість розуміти популярні та прогресивні технології дуже актуальна, особливо зараз, коли все більше з’являється конкурентноспроможних компаній. Засоби та підходи до розробки стають більш інноваційними, а якість розробленого продукту має значну вагу.

Проект «Railroad Tycoon» був виконаний згідно вимог:

* Метод, який у випадковому порядку “розкидуватиме” міста по континенту
* Алгоритм, який знаходитиме точку з найбільшою кучністю
* Алгоритм який знаходить точку з якої варто починати рух
* Алгоритм доєднання нових точок
* Візуалізація процесу

Виконавши відповідні дослідження підходів розробки програмного забезпечення та аналіз можна перейти до постановки задачі на розробку програми. Для початку варто звернути увагу на інтерфейс, дизайн а також стиль майбутньої програми. Важливо вибрати простий і мінімалістичний дизайн з невеликою палітрою кольорів для цього підходить бібліотека matplotlib.

Для конфігурації додатку створенний файл settings.py у якому користувач змінює налаштування програми такі як:

* Ширина і висота вікна
* Кількість міст на площі вікна
* Бюджет
* Розмір області з найбільшою кучністю
* Налаштування кольорів

Для основного функціоналу створюється клас Point який містить всі потрібні методи для задоволення потреб замовника

Файл plot.py який містить функціонал для відображення потрібних даних

Файл main.py який виконує всі потрібні дії для того щоб додаток працював

# РОЗДІЛ 3

## ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДІВ І ЗАСОБІВ РОЗВ’ЯЗАННЯ ЗАВДАННЯ

Основною технологією для розробки програмного забезпечення є Python

### 3.1 Python

Python проста у використанні, та водночас повноцінна мова програмування, що надає набагато більше засобів для структурування і підтримки великих програм, ніж shell. З іншого боку, вона краще за С обробляє помилки, і, будучи мовою дуже високого рівня, має вбудовані типи даних високого рівня, такі як гнучкі масиви і словники, ефективна реалізація яких на C потребує значних витрат часу.

Завдяки більш загальним типам даних, Python застосовують до більш широкого кола задач, ніж Awk і навіть Perl, у той ж час багато речей на мові Python робляться настільки ж просто.

Python дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. Python поставляється з великою бібліотекою стандартних модулів, які можна використовувати як основу для нових програм або як приклади при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережними з'єднаннями і навіть інтерфейсами до різних графічних бібліотек.

Python - інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці “знизу-вверх”. Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми. Програми, написані мовою Python, звичайно значно коротші еквівалента на C або C++ з декількох причин:

* типи даних високого рівня дозволять Вам виразити складні операції однією інструкцією;
* групування інструкцій виконується за допомогою відступів замість фігурних дужок;
* немає необхідності в оголошенні змінних;

Python розширювана мова: знання C дозволяє додавати нові функції, що вбудовуються, або модулі для виконання критичних операцій з максимальною швидкістю або написання інтерфейсу до комерційних бібліотек, доступним тільки у двійковій формі. Інтерпретатор мови Python може бути вбудований у програму, написану на C, і використовувати його як розширення або командну мову для цієї програми. Python використовується в даний час десятками тисяч програмістів в усьому світі, і число людей, що використовують його, швидко зростає, подвоюється і потроюється щороку. Python приваблює користувачів з ряду причин. Він використовується для розробки програм і дозволяє провести розробку набагато швидше, ніж традиційні мови типу C, C++ або Java . Ця мова працює однаково добре на Windows, UNIX, Macintosh, і OS/2, може використовуватися, для легкої розробки як малих додатків чи сценаріїв, так і для розгортання великих програм. Python пропонує доступ до могутнього і легкого у використанні комплекту 29 інструментальних засобів графічного інтерфейсу користувача. Традиційні машинні мови типу C і Pascal мають ряд характеристик, наприклад, сувора типізація, базові типи, складні (і звичайно довгі) цикли, і потреба у великих кількостях кодів для виконання відносно малих задач. Java досить новий, але розділяє більшість характеристик, включених у цей перелік. Програмісти, знайомі з традиційними мовами погодяться, що відсутність суворої типізації полегшує роботу з Python.

Python у порівнянні з іншими мовами

Відмінностей Python від інших мов доволі багато, перерахуємо основні з них:

* Керування пам'яттю - цілком автоматичне — не потрібно хвилюватися щодо розподілу або звільнення пам'яті. Немає загрози “небезпечного посилання”. Java - єдина мова, що пропонує таку концепцію.
* Типи зв'язані з об'єктами, а не зі змінними. Це означає, що змінній може бути призначене значення будь-якого типу, і що (наприклад) масив може містити об'єкти різних типів. Традиційні мови не надають такої можливості.
* Операції звичайно виконуються в більш високому рівні абстракції. Це частково результат того, як написана мова, і частково результат розширеної стандартної бібліотеки кодів, що поставляється разом з Python.

Ці та інші особливості Python роблять розгортання додатків надзвичайно швидким. Продуктивність створеного додатку залежить від його особливостей. Звичайно, для чисельного алгоритму, що виконує звичайну арифметику цілого числа в циклі 'for', неважливо, на якій мові він написаний. Але для “середнього” додатка, збільшення продуктивності може бути просто дивовижним. Один недолік Python, у порівнянні з найбільш традиційними мовами, полягає в тому, що це - не цілком компільована мова; замість цього, вона частково транслює програму до внутрішньої форми байт-коду, і цей байт-код виконується інтерпретатором Python. Однак, у перспективі – сучасні комп'ютери мають так багато невикористовуваного обчислювального потенціалу, що для 90% додатків швидкодія зв'язана з вибором мови. Java теж компілюється в байт-код, але в даний час працює повільніше ніж Python у більшості випадків. Крім того, дуже просто об'єднати Python з модулями, написаними на C або C++, які можна використовувати, щоб збільшити швидкість роботи програм в критичних ділянках.

### 3.2 Numpy

Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник Numpy, Numeric, був спочатку створений Jim Hugunin. Numpy — відкрите програмне забезпечення і має багато розробників.

Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C. NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

### 3.3 Matplotlib

1. matplotlib — бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою (3D графіка також підтримується). Отримувані зображення можуть бути використані як ілюстрації в публікаціях. Зображення, які генеруються в різних форматах, можуть бути використані в інтерактивній графіці, наукових публікаціях, графічному інтерфейсі користувача, веб-додатках, де потрібно будувати діаграми. Бібліотека Matplotlib побудована на принципах ООП, але має процедурний інтерфейс pylab, який надає аналоги команд MATLAB

Matplotlib є гнучким, легко конфігурованим пакетом, який разом з [NumPy](https://uk.wikipedia.org/wiki/NumPy), [SciPy](https://uk.wikipedia.org/wiki/SciPy) і [IPython](https://uk.wikipedia.org/wiki/IPython) надає можливості, подібні до MATLAB. В даний час пакет працює з декількома графічними бібліотеками, включаючи [wxWindows](https://uk.wikipedia.org/wiki/WxWidgets) і [PyGTK](https://uk.wikipedia.org/wiki/PyGTK).

Пакет підтримує багато видів графіків і [діаграм](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0):

* Графіки (line plot)
* [Діаграми розсіювання](https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D1%87%D0%BA%D0%BE%D0%B2%D0%B0_%D0%B4%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0) (scatter plot)
* Стовпчасті діаграми (bar chart) і [гістограми](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D1%81%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0) (histogram)
* [Секторні діаграми](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0#%D0%A1%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D1%96_%D0%B4%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B8) (pie chart)
* [Діаграми «Стовбур-листя»](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%E2%80%9C%D1%81%D1%82%D0%BE%D0%B2%D0%B1%D1%83%D1%80_-_%D0%BB%D0%B8%D1%81%D1%82%D1%8F%E2%80%9D) (stem plot)
* [Контурні графіки](https://uk.wikipedia.org/wiki/%D0%86%D0%B7%D0%BE%D0%BB%D1%96%D0%BD%D1%96%D1%8F) (contour plot)
* Поля градієнтів (quiver)
* [Спектральні діаграми](https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0) (spectrogram)

Користувач може вказати осі координат, сітку, додати підписи і пояснення, використовувати [логарифмічну шкалу](https://uk.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D1%88%D0%BA%D0%B0%D0%BB%D0%B0) або [полярні координати](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8F%D1%80%D0%BD%D1%96_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82%D0%B8).

З допомогою Matplotlib можна створювати і [анімовані](https://uk.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%BB%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F) зображення.

# РОЗДІЛ 4

## ОПИС РЕАЛІЗАЦІЇ ЗАВДАННЯ

Проект Railroad Tycoon був виконаний у визначений термін та виконує поставлену задачу.

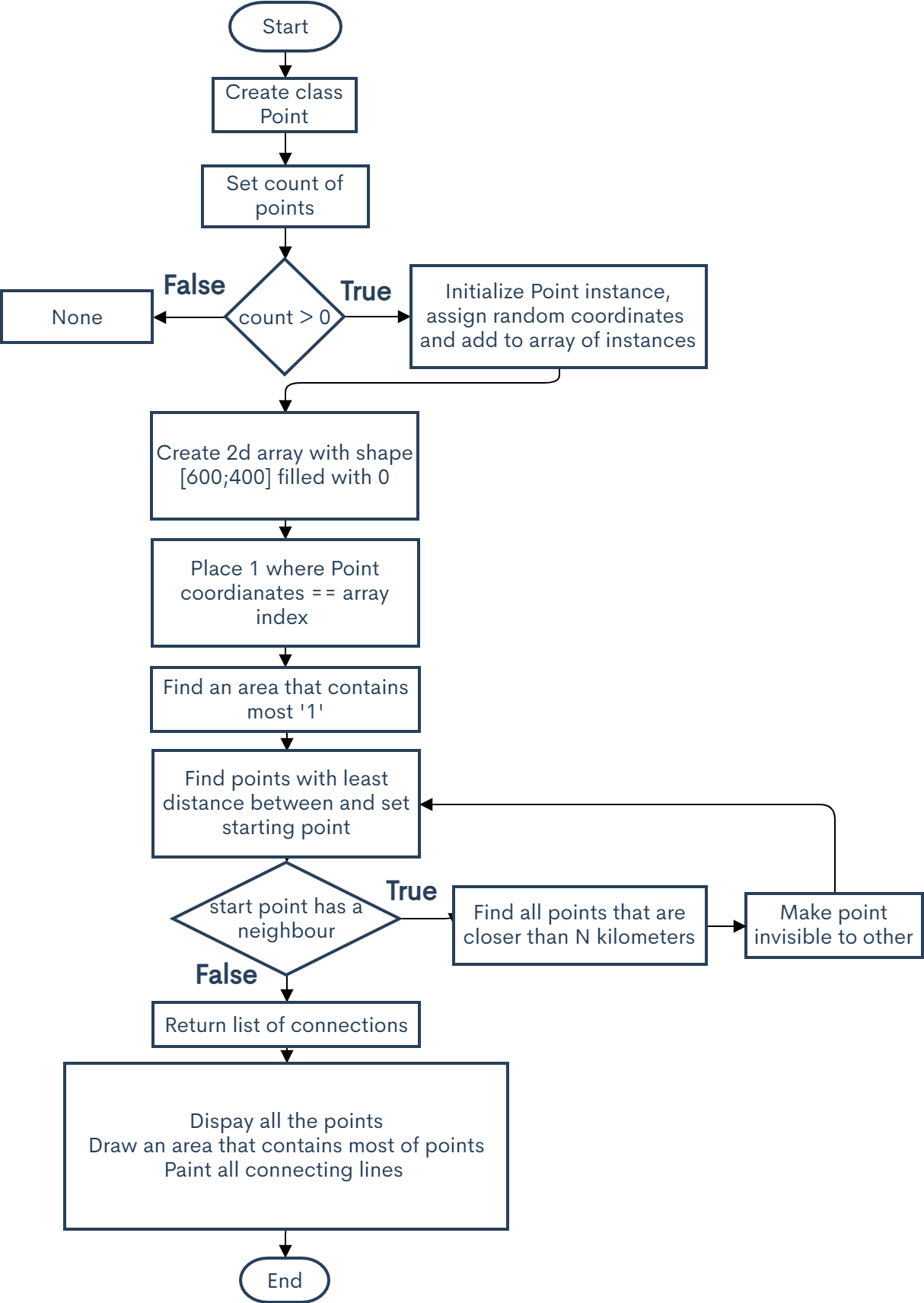
Програма розроблялась на платформі Windows 10 та мові програмування Python.

Для полегшення та оптимізації роботи з даними використовувалась бібліотека Numpy. Це розширення надає ефективніший інструмент для роботи із великими та/або багатомірними масивами даних

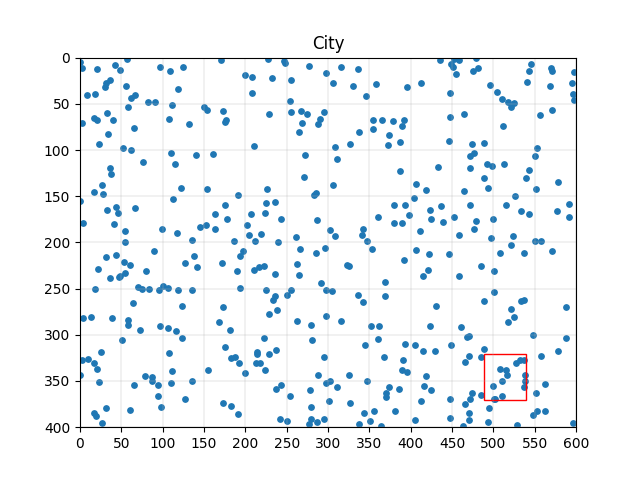
За допомогою бібліотеки scipy відшукувалась відстань між двома містами

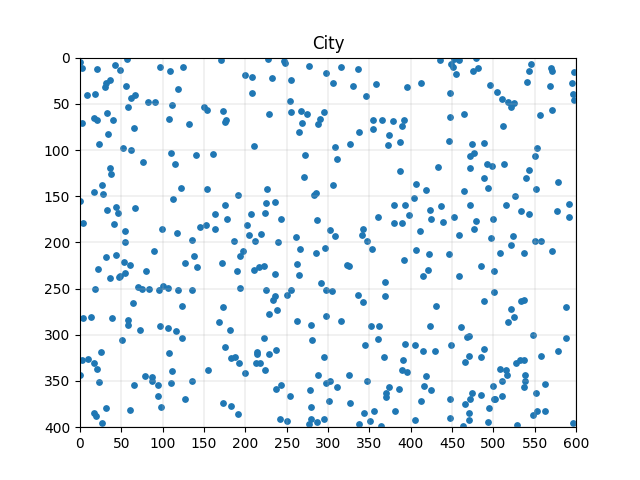
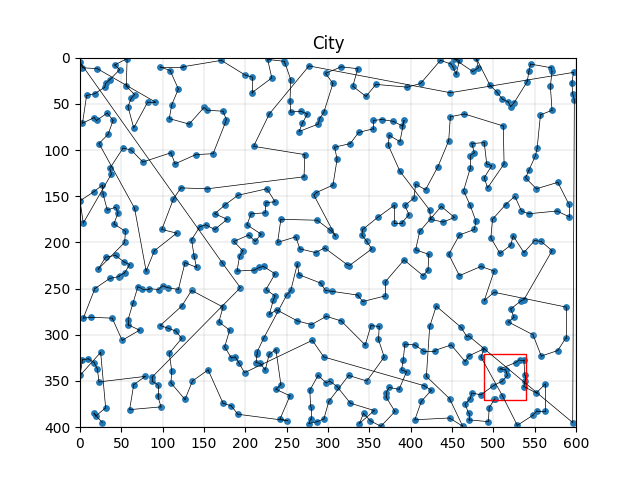
Для візуалізації використовувалась бібліотека matplotlib. Ця бібліотека надає зрозумілий інтерфейс, який добре підходить для цієї задачі.

Вихідний код: https://github.com/AlphazCode/Practice\_Railroad\_Tycoon



## Приклад роботи:





# ВИСНОВКИ

Отже, у процесі проходження практики, було освоєно нові методи дослідження, та розробки програмного забезпечення, вивчено традиційні технології та нові технології для розробки і на основі цього створено алгоритм пошуку оптимального маршруту. Проведення практики включало: обрання теми проекту для практики, вивчення отриманого матеріалу, виконання і перевірка індивідуального проекту а також на завершенні практики презентація виконаного проекту з відповідною темою.

В ході проходження практики зроблено висновок про доцільність розробки даного програмного продукту. Розробка програми завершилася у відповідний термін з виконанням усіх вимог замовника за допомогою сучасних технологій.

# ПЕРЕЛІК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Python:

* https://uk.wikipedia.org/wiki/Python
* https://www.python.org/doc/
* https://www.w3schools.com/python/default.asp

1. Numpy:

* https://numpy.org/
* https://www.w3schools.com/python/numpy/numpy\_intro.asp

1. Matplotlib:

* https://matplotlib.org/
* https://www.w3schools.com/python/matplotlib\_pyplot.asp

1. Все решта: https://www.google.com.ua/