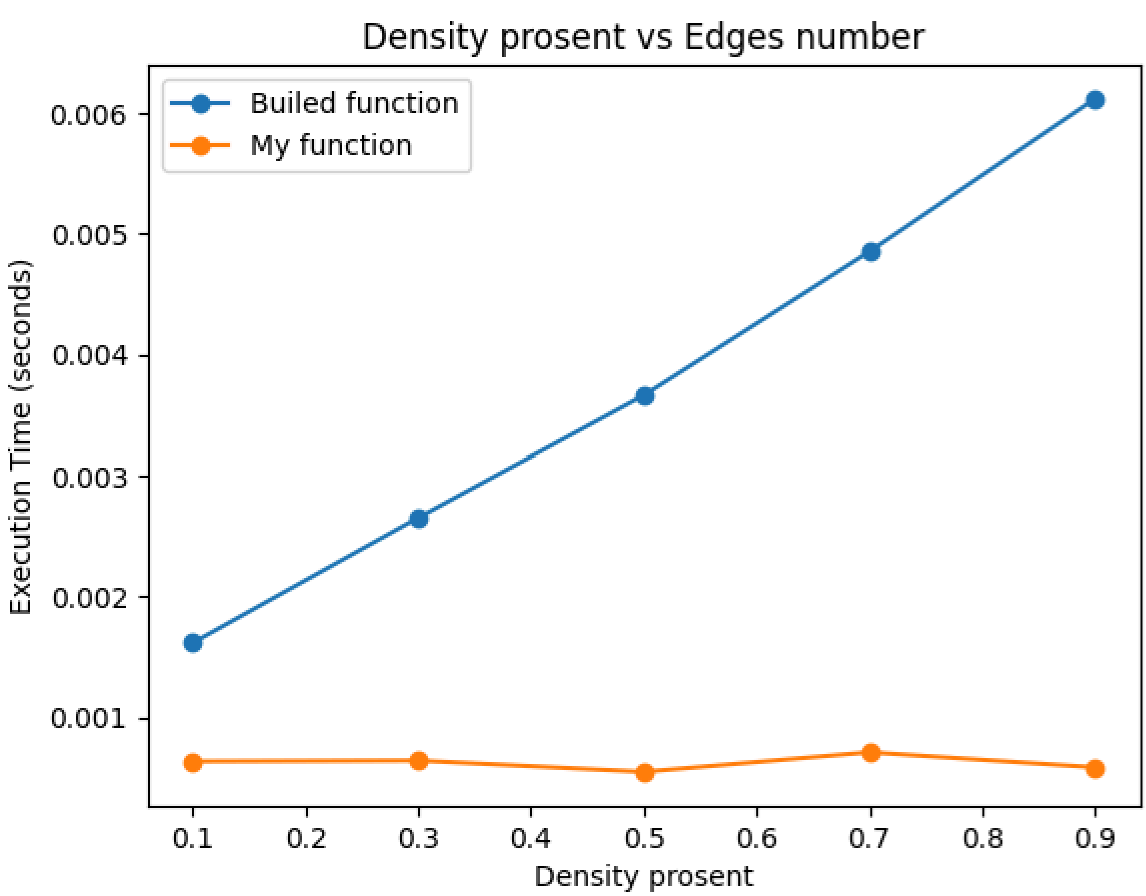
Опис алгоритму Крускала:
Команда імплементувала алгоритм Крускала для знаходження каркасу мінімальної ваги.
Алгоритм Крускала - це жадібний алгоритм, який знаходить мінімальне остовне дерево у зваженому зв'язаному графі.
Наша імпламентація складається з двох додаткових функцій та однієї основної.:

- got_my_implement_graph: Ця функція перетворює дані згенерованого графа у зручний для команди формат. Вона конвертує ребра графа у зручний формат, що містить інформацію про номери вершин та вагу кожного ребра.
- chcose_next_edge: Ця функція вибирає наступне ребро для додавання до каркасу, використовуючи мінімальну вагу серед доступних ребер.
- kruskal_algo: Основна функція, яка використовує попередні дві для знаходження каркасу мінімальної ваги. Вона формує загальний перелік ребер каркасу та обчислює його вагу.

Дослідження ефективності:
Для порівняння ефективності нашої та вбудованої імплементації ми вирішили скористатись графіками.
Досліджували ми два аспекти, котрі можуть впливати:

1) “Насиченість” графа за сталої кількости вершин

density_pros = [0.1, 0.3, 0.5, 0.7, 0.9]
[0.0016167752742767335, 0.002651294708251953, 0.0036605134010314943, 0.0048510880470275875, 0.0061164710521698]
[0.0006353442668914795, 0.0006416175365447998, 0.0005480515956878662, 0.0007101819515228272, 0.0005852525234222412]
Можемо помітити майже пряму пропорцію при підвищенні “насичености” графа



2) Кількість вершин за сталої “насичености”

edges_number = [10, 50, 75, 100, 200]

[0.00010879540443420411, 0.0010099551677703858, 0.004042741060256958, 0.011325592041015626, 0.05430391883850098]

[0.000130403995513916, 0.00019278645515441895, 0.0007003073692321778, 0.0006033937931060791, 0.0007742395401000977]

У цьому випадку можемо спостерігати схожість до параболи