Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра інформатики

Звіт з Практичного завдання №2

По предмету: «Штучний інтелект для інформаційних технологій»

Виконав:

студент групи ІТІНФ-20-1

Самченко С. О.

Харків 2023

**Мета**: ознайомитись на практиці з побудовою агентів, що вирішують задачі, зокрема розглянути проблему навігації / проблему пошуку маршруту.

1. Створення карти:

ukraine\_map = UndirectedGraph(dict(

    Kharkiv =dict(Chuguev =38, Pesochin =11, Kyiv =411),

    Kyiv=dict(Kharkiv =411, Sumy =308, Chernihiv =128, Borispol =33),

    Dnepropetrovsk =dict(Pavlograd =62, Zaporozhye =70, Nikolaev =286),

    Zaporozhye =dict(Melitopol =112),

    Kherson =dict(Nikolaev =59),

    Nikolaev =dict(Odessa =110),

    Donetsk =dict(Lugansk =126),

    Chernihiv =dict(Kharkiv =379, Sumy =251),

    Poltava =dict(Lugansk =365, Kremenchug =100),

    Odessa =dict(Dnepropetrovsk =398, Kyiv =443),

    Lugansk =dict(Kharkiv =272),

    Mirgorod =dict(Poltava =79),

    Berdyansk =dict(Donetsk =157)))

ukraine\_map.locations = dict(

    Kharkiv =(1800, 2500), Kyiv=(1500, 2500), Dnepropetrovsk=(1700, 2400),

    Zaporozhye=(1750, 2350), Kherson=(1600, 2300), Nikolaev=(1550, 2300),

    Odessa=(1500, 2300), Donetsk=(1850, 2400), Lugansk=(1950, 2400),

    Chernihiv=(1550, 2550), Sumy=(1700, 2500), Poltava=(1700, 2450),

    Chuguev=(1800, 2450), Pavlograd=(1750, 2400), Melitopol=(1750, 2300),

    Pesochin=(1800, 2450), Mirgorod=(1650, 2450), Borispol=(1500, 2500),

    Kremenchug=(1650, 2450), Berdyansk=(1800, 2300))

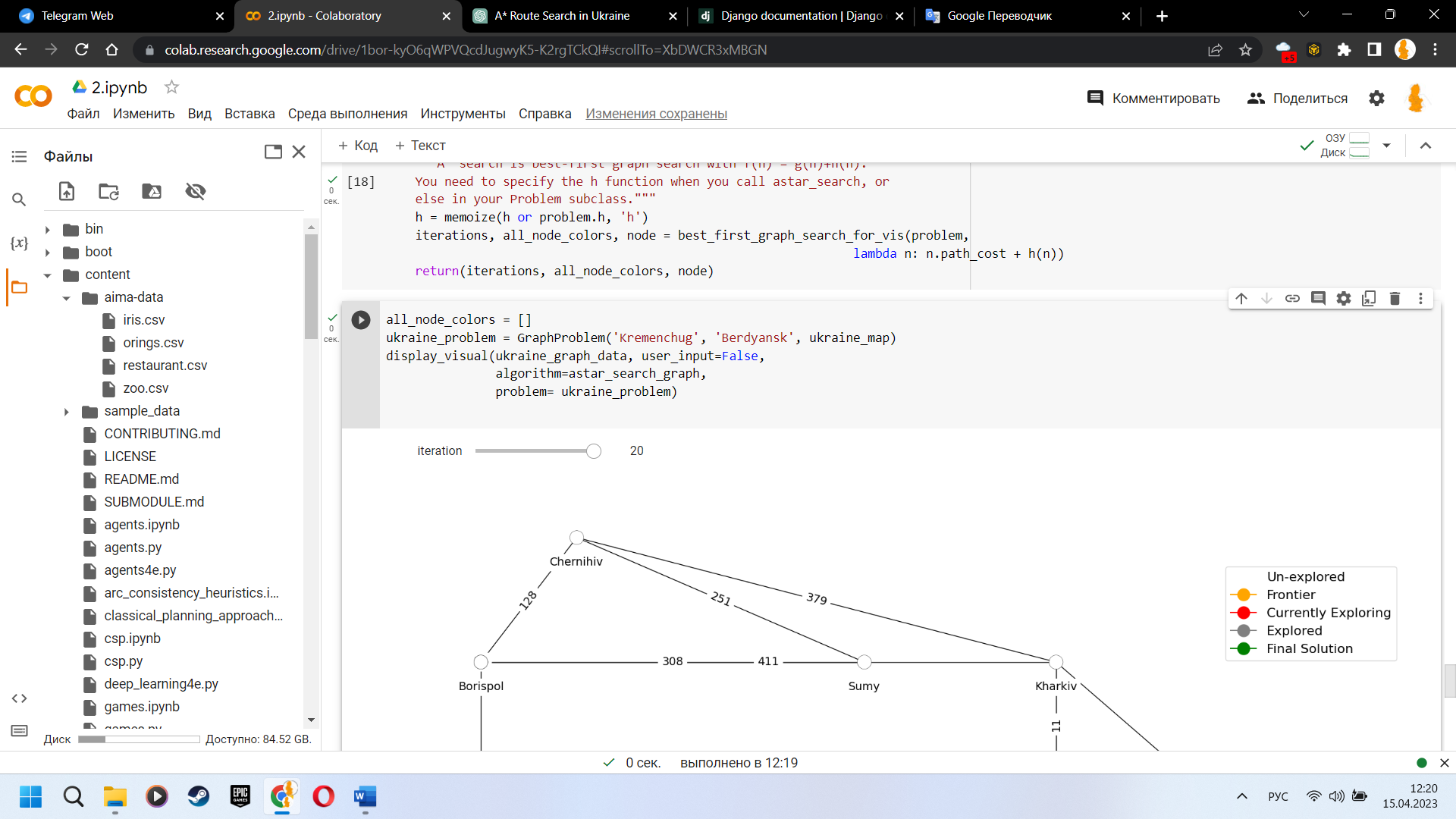


Рис. 1 – Завдання по навігації від Кременчука до Бердянська

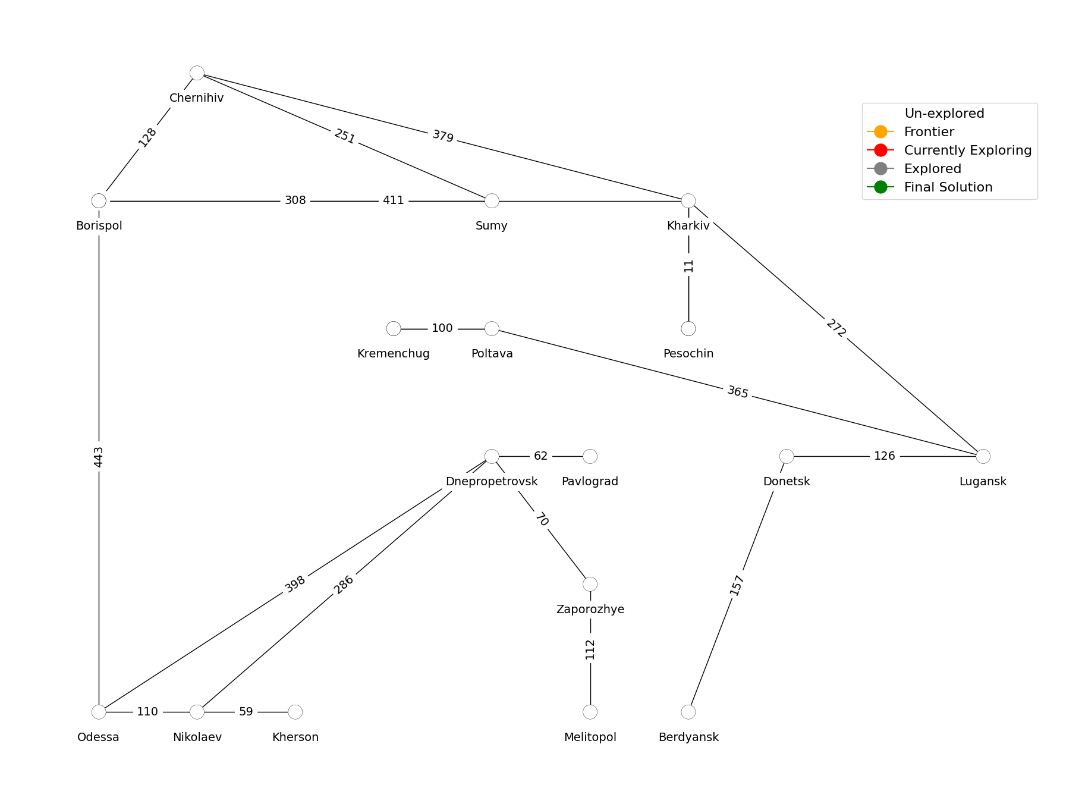


Рис. 2 – відображення мапи

1. Функція, що реалізує алгоритм пошуку:

def astar\_search\_graph(problem, h=None):

    """A\* search is best-first graph search with f(n) = g(n)+h(n).

    You need to specify the h function when you call astar\_search, or

    else in your Problem subclass."""

    h = memoize(h or problem.h, 'h')

    iterations, all\_node\_colors, node = best\_first\_graph\_search\_for\_vis(problem,

                                                                lambda n: n.path\_cost + h(n))

    return(iterations, all\_node\_colors, node)

1. Запуск моделювання пошуку вирішення задачі:

all\_node\_colors = []

ukraine\_problem = GraphProblem('Kremenchug', 'Berdyansk', ukraine\_map)

display\_visual(ukraine\_graph\_data, user\_input=False,

               algorithm=astar\_search\_graph,

               problem= ukraine\_problem)

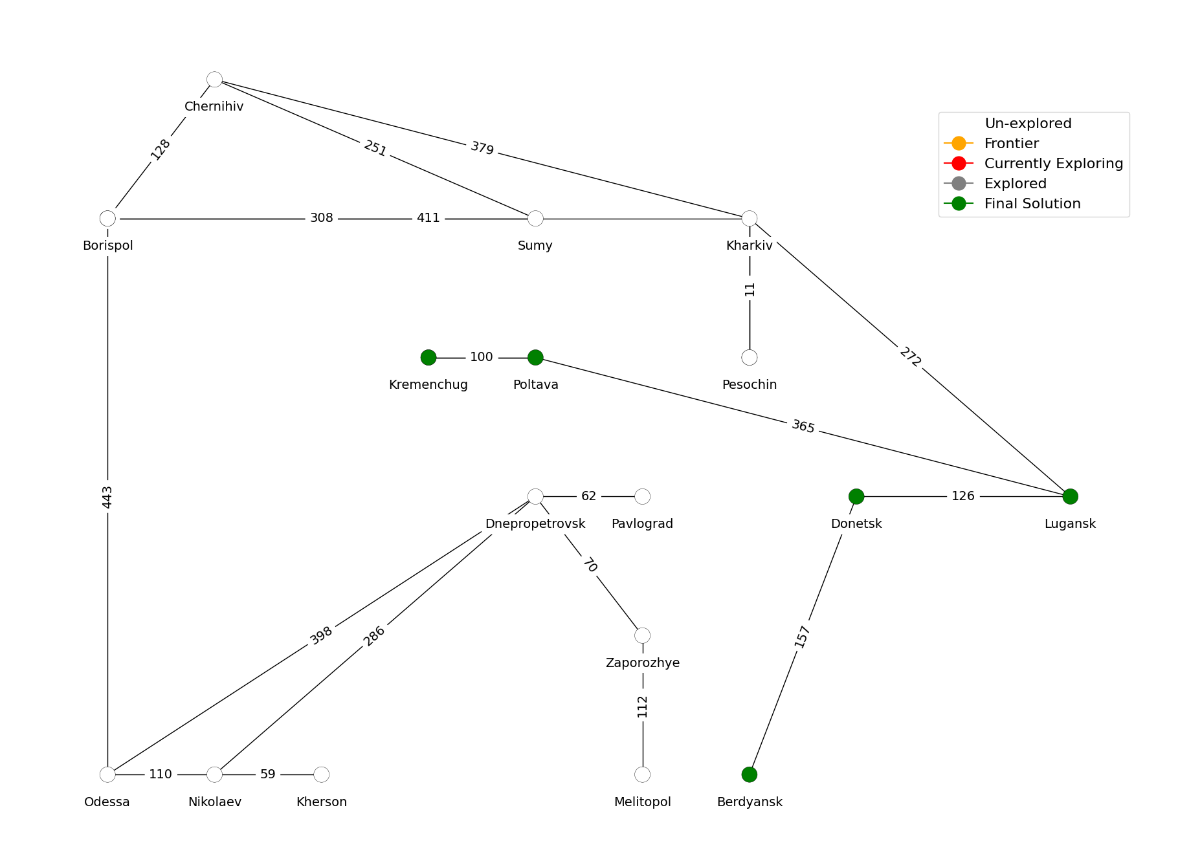


Рис. 3 – маршрут

**Висновок:** у рамках доної роботи були здобуті навички побудови агентів, що вирішують задачі, також були розглянуті проблему навігації / проблеми пошуку маршруту.