МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інформаційних технологій управління

Звіт з лабораторної роботи № 7

з дисципліни «Алгоритми та структури даних»

Виконав:

студент гр. КН-420а

Прохоров Г. В.

Перевірив:

асистент каф. ПІІТУ

Ольховий О. М.

Харків

2021

ТЕМА: ГЕОМЕТРИЧНІ АЛГОРИТМИ

ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Розробити програму, яка читає з клавіатури число N (1 < N < 256) та N пар дійсних чисел — координати точок на площині. Програма виконує один за алгоритмів згідно варіанту.

Варіант 5:

Побудувати опуклу оболонку наданих точок алгоритмом Джарвіса.

МЕТА РОБОТИ

Мета роботи: познайомитися із основними геометричними алгоритмами.

1 ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Основний засіб багатьох геометричних алгоритмів - поняття векторного добутку.

Нехай дано вектора 𝑝1і 𝑝2. Нас цікавлять тільки вектора, що лежать в одній площині, тому під векторним добутком 𝑝1×𝑝2можна розуміти площа паралелограма (з урахуванням знаку), утвореного точками (0;0),𝑝1,𝑝2,𝑝1+𝑝2=(𝑥1+𝑥2;𝑦1+𝑦2). Для обчислень більш зручно визначення векторного добутку як визначника матриці 𝑝1×𝑝2=det(𝑥1𝑥2𝑦1𝑦2)=𝑥1𝑦2—𝑥2𝑦1=−𝑝2×𝑝1.

Якщо 𝑝1×𝑝2позитивне, то найкоротший поворот 𝑝1відносно (0, 0), що поєднує його з 𝑝2, відбувається проти годинникової стрілки, а якщо негативне, то за нею.

Алгоритми побудови опуклої оболонки розглянуто у лекційному курсі, а також у пропонованій літературі.

2 ОПИСАННЯ РОЗРОБЛЕНОЇ ПРОГРАМИ

Весь код програми реалізований у одному файлі реалізації lab7.py. У цьому файлі розв’язуються всі необхідні задачі відповідно до завдання.

Програмний код:

from math import sqrt

import matplotlib.pyplot as plt

dot = []

def main():

N = 0

while True:

try:

N = int(input("Enter N - number of elements: "))

except ValueError:

print("N must be integer")

N = -888 #недопустимое значение

if N >= 2 and N < 256:

break

else:

print("1 < N < 256")

for i in range(N):

tmp = []

print("Dot " + str(i + 1) + ":")

tmp.append(float(input("\tX: ")))

tmp.append(float(input("\tY: ")))

dot.append(tmp)

result = JarvisAlgorithm()

x = []

y =[]

for i in result:

x.append(i[0])

y.append(i[1])

plt.plot(x, y, 'go-', color='blue', linewidth=2)

x = []

y =[]

for i in dot:

x.append(i[0])

y.append(i[1])

plt.plot(x, y, 'ro', markersize=8)

plt.show()

def vectorProduct(first, second, p):

a = [first[0] - p[0], first[1] - p[1]]

b = [second[0] - p[0], second[1] - p[1]]

return a[0] \* b[1] - b[0] \* a[1]

def vectorLen(first, second):

a = [first[0] - second[0], first[1] - second[1]]

return sqrt(a[0] \*\* 2 + a[1] \*\* 2)

def JarvisAlgorithm():

Q = []

Q.append(findMinimum())

while True:

p = Q[len(Q) - 1]

best = dot[0]

if best == p:

best = dot[1]

line = []

for i in dot[1:]:

if p == i:

continue

res = vectorProduct(best, i, p)

if res < 0:

line = []

best = i

elif res == 0:

a = vectorLen(best, p)

b = vectorLen(i, p)

if (b > a):

line.append(best)

best = i

if len(line) != 0:

for i in line:

Q.append(i)

Q.append(best)

if best == Q[0]:

break

return Q

def findMinimum():

minimum = dot[0]

for i in dot[1:]:

if i[1] < minimum[1]:

minimum = i

elif i[1] == minimum[1]:

if i[0] < minimum[0]:

minimum = i

return minimum

main()

3 РЕЗУЛЬТАТИ

Розглянемо ситуацію, коли всі елементи вводяться коректно згідно з умовою завдання (рис. 3.1, рис 3.2).

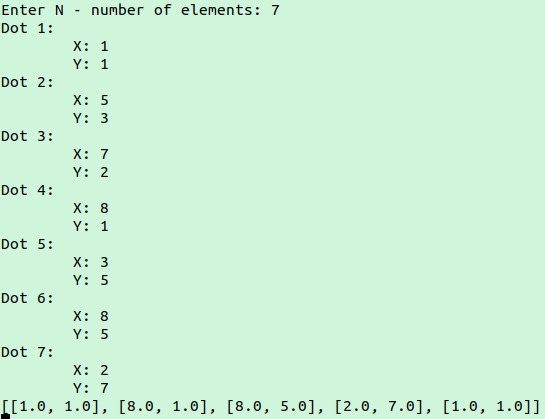


Рисунок 3.1 – Вводяться коректні дані

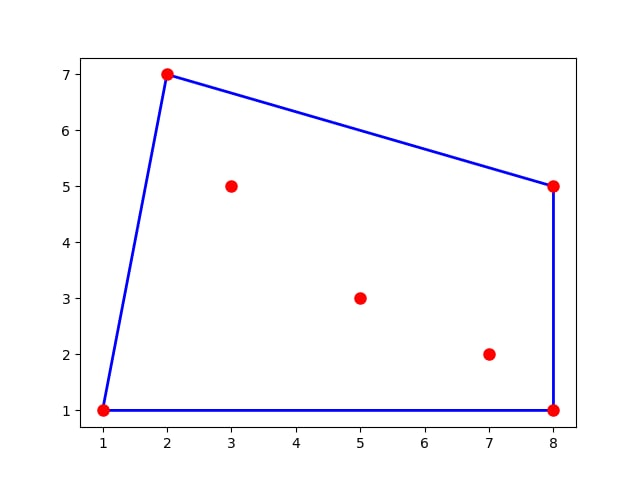


Рисунок 3.2 – Графік при вводі коректних даних

При введенні кількості елементів N, що не задовольняє інтервал, програма, - попереджає про це й пропонує перезаписати N.  (рис. 3.3).

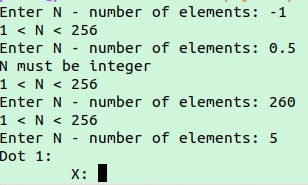


Рисунок 3.3 – Робота програми при введенні некоректного N

ВИСНОВКИ

За допомогою цієї роботи були отримані нові знання, пов’язані із основними геометричними алгоритмами. Реалізований у програмі код повністю відповідає умові задачі. Опукла оболонка наданих точок розраховувалась алгоритмом Джарвіса. Результати виводились графіком на площині. Для обробки інформації було зроблено декілька функцій, кожна з яких виконує свою задачу.

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1 Конспект лекцій з дисципліни «Основи теорії алгоритмів».

2 Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритми побудови та аналіз: підручник / Китаєв А.Ю., Щень А., Вялий М.Н.: Видавництво «МЦНМО», 2002. – 1398 с.

3 Д. Е. Кнут. Мистецтво програмування: підручник / Штонда В., Петріковец Г., Орловіч О., 1990. – 682 с.

4 А. В. Ахо. Структури даних й алгоритми / А. В. Ахо, Д.Э. Хопкрофт, Д. Д. Ульман. – М.: Вильямс, 2003. – 384 с.