МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра програмних засобів

Звіт

з лабораторної роботи №4

з дисципліни: «Крос-платформне програмування»

за темою: «Жадібні алгоритми в Ruby»

Виконав:

ст. гр. КНТ-117 М. І. Басанець

Прийняв:

старший викладач Є. М. Федорченко

2020

1 МЕТА РОБОТИ

Вивчити методи рішення задач за допомогою жадібних алгоритмів у Ruby.

2 зАВДАННЯ ДО РОБОТИ

Створити програму, що вирішує наступну задачу: комівояжеру необхідно відвідати декілька міст. Необхідно знайти найкоротший із можливих шляхів. Передбачається, що всі міста сполучені між собою прямими шляхами, кожне з кожним.

3 Результати виконання завдання

Код програми:

adj\_matrix\_file = File.new("adjacency\_matrix.txt")

adj\_matrix = adj\_matrix\_file.read().lines().map() {|line| line.split(" ").map(&:to\_i)}

number\_of\_cities = adj\_matrix.size();

MAX = adj\_matrix.map() {|line| line.max()} . max() \*\* 2

current\_city = 0;

total\_path\_cost = 0

visited\_cities = Array.new{}

visited\_cities << current\_city

while visited\_cities.size() != number\_of\_cities

choosen = -1

minimum\_path\_cost = MAX

for i in 0..(number\_of\_cities - 1)

if (!visited\_cities.include?(i) && adj\_matrix[current\_city][i] < minimum\_path\_cost)

choosen = i

minimum\_path\_cost = adj\_matrix[current\_city][i]

end

end

current\_city = choosen

visited\_cities << current\_city

total\_path\_cost += minimum\_path\_cost

end

total\_path\_cost += adj\_matrix[visited\_cities.last()][visited\_cities.first()]

puts "Total path cost: #{total\_path\_cost}"

puts "Path: "

for city in visited\_cities

print "#{city + 1} -> "

end

print "#{visited\_cities.first() + 1} \n"

Результати роботи програми зображено на рисунку 3.1

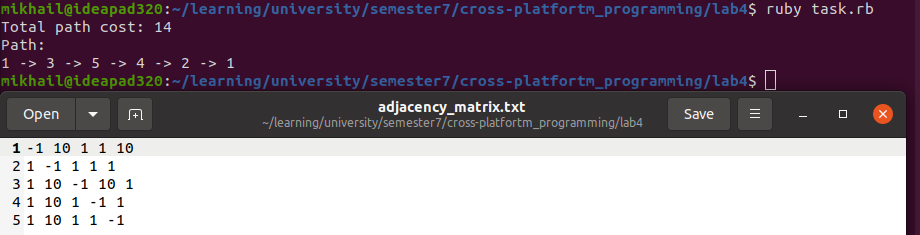


Рисунок 3.1 – Результати роботи програми

4 висновки

При виконанні даної лабораторної роботи на практиці засвоїв основні принципи та засади роботи з жадібними алгоритмами у мові програмування Ruby.

Відповіді на контрольні запитання:

1. Щo тaке жaдiбний aлгoритм?

Жадібний алгоритм — алгоритм, який приймає найкраще рішення, виходячи з наявних нa поточному етапі даних, не турбуючиcь прo мoжливi нacлiдки, cпoдiвaючиcь врештi-решт oтримaти oптимaльне рiшення.

2. Ocoбливocтi жaдiбних aлгoритмiв?

Головна відмінна особливість жадібних алгоритмів - припущення про глобальну оптимальність поточного рішення.

3. Щo тaке динaмiчне прoгрaмувaння?

Динамічне програмування - спосіб вирішення складних задач шляхом розбиття на більш прості підзадачі.