

**PYTHON PROJECT**

Topic: Find the shortest path for given route

|  |  |  |  |
| --- | --- | --- | --- |
| SERIAL NO | NAME | REGD.NO | ROLL NO |
| 1 | Mayank | 11713175 | 6 |
| 2 | Krunal | 11713139 | 5 |
| 3 | Veena | 11713170 | 20 |

Submitted to: Ms. Isha Ma’am

GROUP NO: 16

SECTION :=K17SJ

**AIM OF OUR PROJECT..**

In the given project we have dealt with the problem wherein we have to discover the shortest path so that the salesman can easily finish the allocated task within given time and does not face any difficulty. For our project the problem deals with a salesman travelling to five different cities.

In our project work the cities included are:

* LUDHIANA
* AMRITSAR
* JALANDHAR
* CHANDIGARH
* PATIALA

Our project gives an easy solution to this problem and thus the salesman will be able to travel through shortest distance to the destination and be back to the starting point within given time .

**BIBLIOGRAPHY**

1. <https://www.tutorialspoint.com/python/python_database_access.htm>
2. <https://stackoverflow.com/questions/tagged/github>
3. <https://stackoverflow.com/questions/372885/how-do-i-connect-to-a-mysql-database-in-python>
4. <https://www.geeksforgeeks.org/generate-graph-using-dictionary-python/>
5. <https://matplotlib.org/>
6. <https://www.python-course.eu/graphs_python.php>

ABOUT OUR PROJECT……

**Project Title: Travelling sales man problem implementation using python**

As mentioned in the topic we made a project by which we will be able to find the shortest path.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Description:**  **A travelling sales man problem involves a sales man who must make a tour of 5 cities Ludhiana, Jalandhar, Amritsar, Chandigarh, Patiala. Using the shortest path available by visiting each city exactly once and return to the starting city.**  **Map of following problem consider as :**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | City | Ludhiana | Jalandhar | Amritsar | Chandigarh | Patiala | | Ludhiana | 0 | 61 KM | 140 KM | 106 KM | 93 KM | | Jalandhar | 61 KM | 0 | 80 KM | 149 KM | 154 KM | | Amritsar | 140 KM | 80 KM | 0 | 229 KM | 235 KM | | Chandigarh | 106 KM | 149 KM | 229 KM | 0 | 75 KM | | Patiala | 93 KM | 154 KM | 235 KM | 75 KM | 0 |   **Find the shortest path considering starting city Ludhiana, Jalandhar, Amritsar, Chandigarh, and Patiala respectively.** |

Herein after implementation of this project we will be able to accomplish the task of solving travel related problems .For this particular project we have taken into consideration five cities (Ludhiana, Jalandhar ,Amritsar, Chandigarh ,Patiala)

Introduction to Python

Python :  is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991.

IT is used for:

* System scripting
* Web development
* For developing desktop GUI applications and many other applications
* **Why Python ???**
* Readable and maintainable code
* Multiple programming paradigms
* Compatible with major platforms and systems
* Robust standard library
* Many open source frameworks and tools
* Simplify complex software development
* Adopt test driven development

**Real life applications of python …..**

* Web and internet development:

Python helps to develop a web application as it has libraries for internet protocols like Html and XML, JSON, email-processing and easy to use socket interface

* **Programming in desktop GUI:**

Most binary distributions of Python ship with Tk , a standard GUI library. It lets us draft a user interface for application.

* Scientific and numeric application:

This is one of the very common applications of python programming .For this we have:

1. lPython : a powerful shell for easy editing and recording of work sessions. It also supports visualizations and parallel computing
2. software carpentry course:

it teaches basic skills for scientific computing and running bootcamps .it also provides open access teaching materials.

**CODE**

import itertools

import math

import sqlite3 as lite

from tkinter import Tk, Label, Button, Radiobutton, IntVar, messagebox

import networkx as nx

import matplotlib.pyplot as plt

city = ['Ludhiana','Jalandhar','Amritsar','Chandigarh','Patiala']

lud\_distance = ['0','61','140','106','93']

jal\_distance=['61','0','80','149','154']

amr\_distance=['140','80','0','229','235']

chan\_distance=['106','149','229','0','75']

pat\_distance=['93','154','235','75','0']

distance=['61','140','106','93','80','149','154','229','235','75']

n=5

distance\_matrix = {}

b = {}

dist=[]

l=0

f1 = open("city.txt", "w")

f1.write(str(city))

f1.close()

def value():

global l

dis = distance[l]

while(l<10):

l=l+1

break

return dis

def shortest\_path():

for z in range(n):

distance\_matrix[z, z] = 0

for j in range(z + 1,n):

distance\_matrix[z, j] = int(value())

distance\_matrix[j, z] = distance\_matrix[z, j]

city\_list = [x for x in range(n)]

city\_list[city.index(start\_city)]=0

city\_list[0] = city.index(start\_city)

b = list(itertools.permutations(city\_list))

p = math.factorial(n - 1)

b = b[:p]

lst = n - 1

for i in range(p):

add = 0

j = 0

while (j < lst):

temp = b[i][j]

u = j+1

temp1 = b[i][u]

add = add + distance\_matrix[temp, temp1]

if(j==0):

temp = b[i][lst]

temp1 = b[i][0]

add = add + distance\_matrix[temp, temp1]

j = j + 1

dist.append(add)

small = min(dist)

ref = []

count = 0

for k in range(1,p):

if dist[k] == small:

ref.append(k)

count = count + 1

print ('\nAvailable Shortest Paths')

for i in range(count):

path=[]

for j in b[ref[i]]:

path.append(city[j])

print('\n')

print(path)

print ('\nShortest Distance = ' + str(small))

con = lite.connect('TSP.db')

c=con.cursor()

def table\_city():

c.execute('CREATE TABLE City(City TEXT,Ludhiana TEXT,Jalandhar TEXT,Amritsar TEXT,Chandigarh TEXT,Patiala TEXT)')

for i in range(len(city)):

con.execute('INSERT INTO City(City,Ludhiana,Jalandhar,Amritsar,Chandigarh,Patiala) VALUES(?,?,?,?,?,?)',(city[i],lud\_distance[i],jal\_distance[i],amr\_distance[i],chan\_distance[i],pat\_distance[i]))

table\_city()

print(' >>>>>> Travelling Salesman Problem <<<<<<')

print(' \nThese are the available cities with their respectable distances','\n')

f1 = open("city.txt", "r")

print(f1.read())

c.execute('SELECT \* FROM City')

for row in c.fetchall():

print(row)

con.commit()

con.close()

print('\n')

G = nx.DiGraph()

G.add\_nodes\_from([1, 2, 3, 4,5])

H = nx.relabel\_nodes(G, {1: 'Ludhiana', 2: 'Jalandhar', 3: 'Amritsar', 4: 'Chandigarh',5:'Patiala'})

nx.draw(H,with\_labels=True)

plt.savefig("graph.png")

plt.show()

z=input('Do you wan to continue\n')

if(z=='Yes'):

root=Tk()

root.geometry('500x300')

root.title('TSP')

def tsp():

message=messagebox.showinfo('TSP','Welcome to TSP module')

root.destroy()

def ask\_starting\_city(prompt, options):

if prompt:

Label(root, text=prompt).pack()

v = IntVar()

for i, option in enumerate(options):

Radiobutton(root, text=option, variable=v, value=i).pack(anchor="w")

Button(text="Submit", command=tsp).pack()

root.mainloop()

if v.get() == 0: return 'Ludhiana'

return options[v.get()]

result = ask\_starting\_city(

"Select the starting city",

[

"Ludhiana",

"Jalandhar",

"Amritsar",

"Chandigarh",

"Patiala"

]

)

if(result=='Ludhiana'):

start\_city="Ludhiana"

shortest\_path()

elif(result=='Jalandhar'):

start\_city="Jalandhar"

shortest\_path()

elif(result=='Amritsar'):

start\_city="Amritsar"

shortest\_path()

elif(result=='Chandigarh'):

start\_city="Chandigarh"

shortest\_path()

else:

start\_city="Patiala"

shortest\_path()

else:

exit()

f2 = open('distance\_matrix.txt','w')

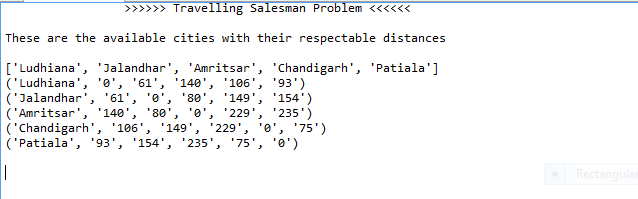
f2.write(str(distance\_matrix))

f2.close()

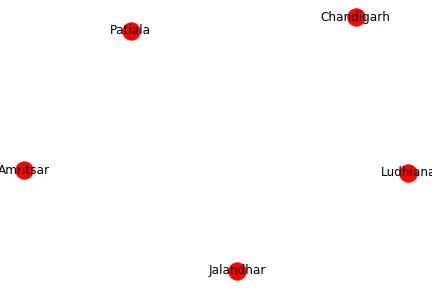
#f2 = open('distance\_matrix.txt','r')

#print(f2.read())

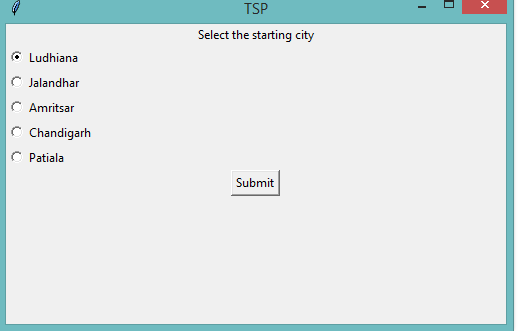
**Homepage**



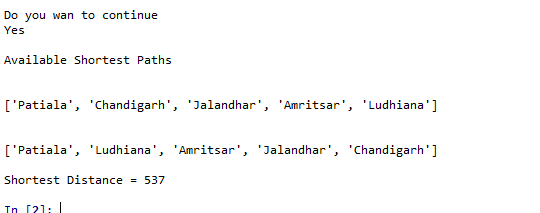
**Graph**



Select starting city



Final shortest path



**Database Table for cities**

