

DATA: 17/01/2022	Title of the Lab	Name: Avinash reddy Vasipalli
EXP No: 02	AGENTS AND REAL - WORLDPROBLEMS	Registration Number: RA1911027010007 Section: N1 Lab Batch: 1 Day Order: 2

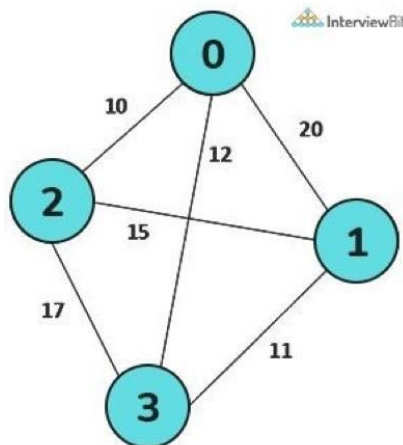
Travelling Salesman Problem

AIM: To Implement Travelling Salesman Problem using python.

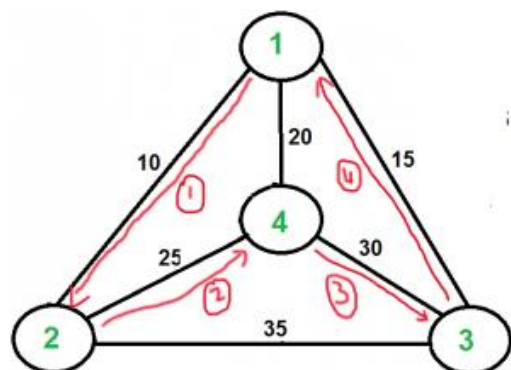
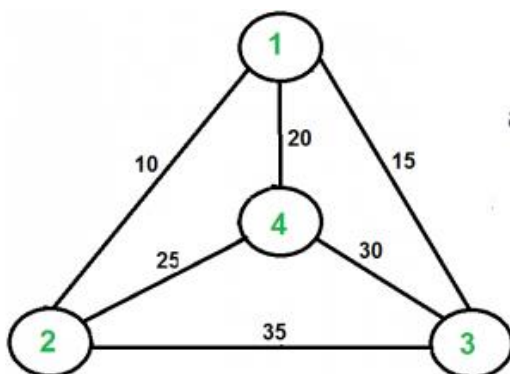
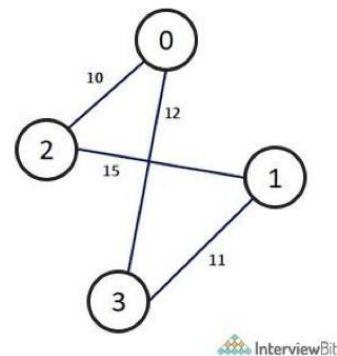
Description of the concept or Problem given

Given set of cities and distance between every pair of cities. The problem is to find shortest possible route that can visit every city exactly once and return to the starting point.

Manual Solution



The Shortest Path
Covering All The Nodes



Problem 1

	0	1	2	3
0	0	20	10	12
1	20	0	15	11
2	10	15	0	17
3	12	11	17	0

Problem 2

	1	2	3	4
1	0	10	15	20
2	10	0	35	25
3	15	35	0	30
4	20	25	30	0

For the 2 problems**From Problem 1**

From node 0 the shortest distance is 10, so move to node 2.

From node 2 the shortest distance is 15, so move to node 1.

From node 1 the shortest distance is 11, so move to node 3.

From node 3 the shortest distance is 12, so move back to node 0.

So, the shortest distance route would be $10+15+11+12 = 48$

From Problem 2

From node 1 the shortest distance is 10, so move to node 2.

From node 2 the shortest distance is 25, so move to node 4.

From node 4 the shortest distance is 30, so move to node 3.

From node 3 the shortest distance is 15, so move back to node 1.

So, the shortest distance route would be $10+25+30+15 = 80$

Program Implementation [Coding]

```
LAB2.py X
C: > Users > Avinash > Desktop > LAB2.py > TSP > current_pathweight
1  from sys import maxsize
2  from itertools import permutations
3  A = 4
4
5  def TSP(graph, s):
6      vertex = []
7      for i in range(A):
8          if i!=s:
9              vertex.append(i)
10
11      min_path = maxsize
12      next_permutation = permutations(vertex)
13      for i in next_permutation:
14          current_pathweight = 0
15          k = s
16          for j in i:
17              current_pathweight += graph[k][j]
18              k=j
19              current_pathweight += graph[k][s]
20
21          min_path = min(min_path, current_pathweight)
22      return min_path
23
24  if __name__ == "__main__":
25      graph = [[0,10,15,20],[10,0,35,25],
26              [15,35,0,30],[20,25,30,0]]
27      s= 0
28      print(TSP(graph, s))
```

```
LAB2.py X
C: > Users > Avinash > Desktop > LAB2.py > ...
1  from sys import maxsize
2  from itertools import permutations
3  A = 4
4
5  def TSP(graph, s):
6      vertex = []
7      for i in range(A):
8          if i!=s:
9              vertex.append(i)
10
11      min_path = maxsize
12      next_permutation = permutations(vertex)
13      for i in next_permutation:
14          current_pathweight = 0
15          k = s
16          for j in i:
17              current_pathweight += graph[k][j]
18              k=j
19              current_pathweight += graph[k][s]
20
21          min_path = min(min_path, current_pathweight)
22      return min_path
23
24  if __name__ == "__main__":
25      graph = [[0,20,10,12],[20,0,15,11],
26              [10,15,0,17],[12,11,17,0]]
27      s= 0
28      print(TSP(graph, s))
```

```
from sys import maxsize
from itertools import permutations
A = 4
```

```
def TSP(graph, s):
    vertex = []
    for i in range(A):
        if i!=s:
            vertex.append(i)

    min_path = maxsize
    next_permutation = permutations(vertex)
    for i in next_permutation:
        current_pathweight = 0
        k = s
        for j in i:
            current_pathweight += graph[k][j]
            k=j
            current_pathweight += graph[k][s]

        min_path = min(min_path, current_pathweight)
    return min_path
```

```
if __name__ == "__main__":
    graph = [[0,20,10,12],[20,0,15,11],
             [10,15,0,17],[12,11,17,0]]
    s= 0
    print(TSP(graph, s))
```

Screenshots of the Outputs

```
LAB2.py X
C: > Users > Avinash > Desktop > LAB2.py > ...
1 from sys import maxsize
2 from itertools import permutations
3 A = 4
4
5 def TSP(graph, s):
6     vertex = []
7     for i in range(A):
8         if i!=s:
9             vertex.append(i)
10
11     min_path = maxsize
12     next_permutation = permutations(vertex)
13     for i in next_permutation:
14         current_pathweight = 0
15         k = s
16         for j in i:
17             current_pathweight += graph[k][j]
18             k=j
19         current_pathweight += graph[k][s]
20
21     min_path = min(min_path, current_pathweight)
22     return min_path
23
24 if __name__ == "__main__":
25     graph = [[0,10,15,20],[10,0,35,25],
26             [15,35,0,30],[20,25,30,0]]
27     s= 0
28     print(TSP(graph, s))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
80 PS C:\Users\Avinash> python -u "c:\Users\Avinash\Desktop\LAB2.py"
80 PS C:\Users\Avinash> python -u "c:\Users\Avinash\Desktop\LAB2.py"
80 PS C:\Users\Avinash>
```

```
LAB2.py X
C: > Users > Avinash > Desktop > LAB2.py > ...
1 from sys import maxsize
2 from itertools import permutations
3 A = 4
4
5 def TSP(graph, s):
6     vertex = []
7     for i in range(A):
8         if i!=s:
9             vertex.append(i)
10
11     min_path = maxsize
12     next_permutation = permutations(vertex)
13     for i in next_permutation:
14         current_pathweight = 0
15         k = s
16         for j in i:
17             current_pathweight += graph[k][j]
18             k=j
19         current_pathweight += graph[k][s]
20
21     min_path = min(min_path, current_pathweight)
22     return min_path
23
24 if __name__ == "__main__":
25     graph = [[0,20,10,12],[20,0,15,11],
26             [10,15,0,17],[12,11,17,0]]
27     s= 0
28     print(TSP(graph, s))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
48 PS C:\Users\Avinash> python -u "c:\Users\Avinash\Desktop\LAB2.py"
48 PS C:\Users\Avinash>
```

Result:

Successfully implemented the code to find the shortest path between cities for the travelling salesman problem.

Medical Diagnosis System

AIM: To Implement Medical Diagnosis System using python.

Description of the concept or problem given:

Medical diagnosis is basically a pattern classification phenomenon: based on some input given by patient and a expert gives a conclusion on the basis of its knowledge. This is stored in binary form, and finally the result is calculated to the point either patient suffering from disease or not.

Manual Solution:

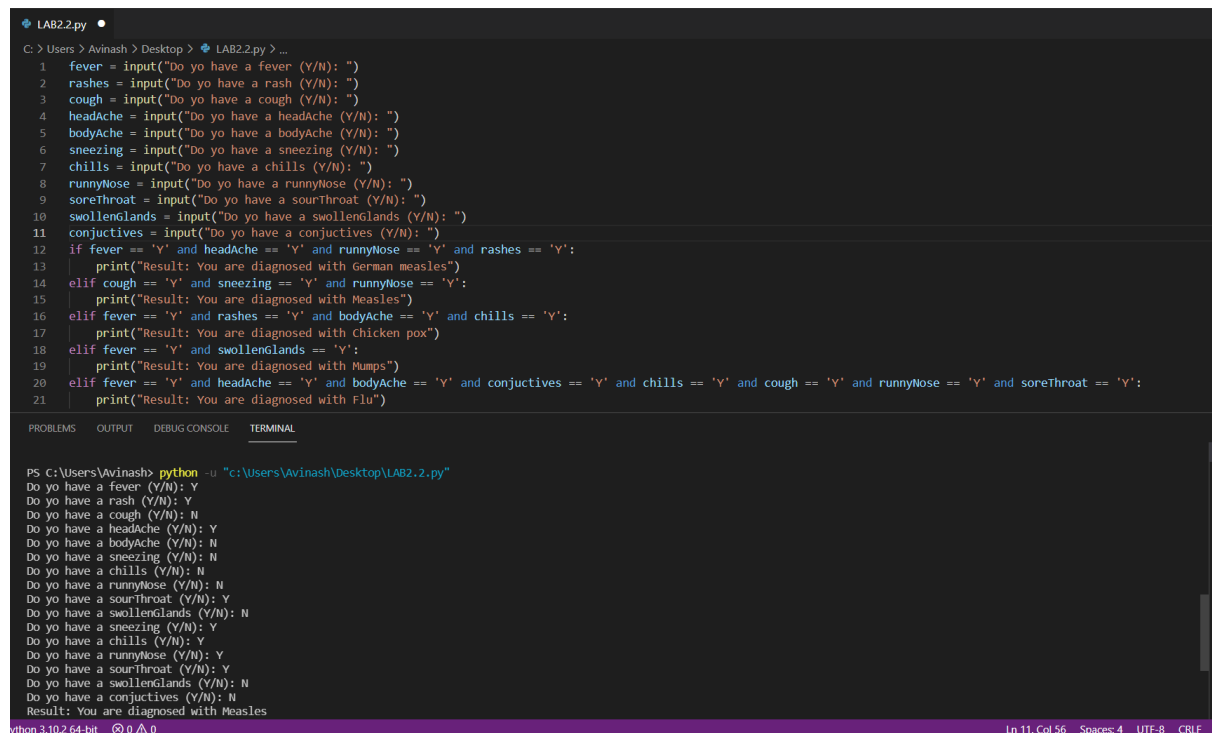
This is basic program in Python where the patient is asked series of questions and for which the patient replies YES or NO. This process is setup with if and elif conditions for every question asked and answers given will confirm the diagnosis.

Program Implementation [Coding]

```
fever = input("Do yo have a fever (Y/N): ")
rashes = input("Do yo have a rash (Y/N): ")
cough = input("Do yo have a cough (Y/N): ")
headAche = input("Do yo have a headAche (Y/N): ")
bodyAche = input("Do yo have a bodyAche (Y/N): ")
sneezing = input("Do yo have a sneezing (Y/N): ")
chills = input("Do yo have a chills (Y/N): ")
runnyNose = input("Do yo have a runnyNose (Y/N): ")
soreThroat = input("Do yo have a sourThroat (Y/N): ")
swollenGlands = input("Do yo have a swollenGlands (Y/N): ")
conjuctives = input("Do yo have a conjuctives (Y/N): ")
```

```
if fever == 'Y' and headAche == 'Y' and runnyNose == 'Y' and rashes == 'Y':
    print("Result: You are diagnosed with German measles")
elif cough == 'Y' and sneezing == 'Y' and runnyNose == 'Y':
    print("Result: You are diagnosed with Measles")
elif fever == 'Y' and rashes == 'Y' and bodyAche == 'Y' and chills == 'Y':
    print("Result: You are diagnosed with Chicken pox")
elif fever == 'Y' and swollenGlands == 'Y':
    print("Result: You are diagnosed with Mumps")
elif fever == 'Y' and headAche == 'Y' and bodyAche == 'Y' and conjuctives == 'Y' and chills == 'Y' and
cough == 'Y' and runnyNose == 'Y' and soreThroat == 'Y':
    print("Result: You are diagnosed with Flu")
```

Screenshots of the Outputs



The screenshot displays a Python IDE with a dark theme. The top pane shows the code for LAB2.2.py, which is a diagnostic program for German measles, Measles, Chicken pox, Mumps, and Flu. The code uses input() to collect user responses for various symptoms and if-elif-else statements to determine the diagnosis. The bottom pane shows the terminal output, where the user has entered 'Y' for fever, rash, headAche, bodyAche, sneezing, chills, runnyNose, and soreThroat, and 'N' for cough, swollenGlands, and conjuctives. The program has identified the diagnosis as Measles.

```
LAB2.2.py
C:\Users\Avinash\Desktop> LAB2.2.py
1 fever = input("Do yo have a fever (Y/N): ")
2 rashes = input("Do yo have a rash (Y/N): ")
3 cough = input("Do yo have a cough (Y/N): ")
4 headAche = input("Do yo have a headAche (Y/N): ")
5 bodyAche = input("Do yo have a bodyAche (Y/N): ")
6 sneezing = input("Do yo have a sneezing (Y/N): ")
7 chills = input("Do yo have a chills (Y/N): ")
8 runnyNose = input("Do yo have a runnyNose (Y/N): ")
9 soreThroat = input("Do yo have a sourThroat (Y/N): ")
10 swollenGlands = input("Do yo have a swollenGlands (Y/N): ")
11 conjuctives = input("Do yo have a conjuctives (Y/N): ")
12 if fever == 'Y' and headAche == 'Y' and runnyNose == 'Y' and rashes == 'Y':
13     print("Result: You are diagnosed with German measles")
14 elif cough == 'Y' and sneezing == 'Y' and runnyNose == 'Y':
15     print("Result: You are diagnosed with Measles")
16 elif fever == 'Y' and rashes == 'Y' and bodyAche == 'Y' and chills == 'Y':
17     print("Result: You are diagnosed with Chicken pox")
18 elif fever == 'Y' and swollenGlands == 'Y':
19     print("Result: You are diagnosed with Mumps")
20 elif fever == 'Y' and headAche == 'Y' and bodyAche == 'Y' and conjuctives == 'Y' and chills == 'Y' and cough == 'Y' and runnyNose == 'Y' and soreThroat == 'Y':
21     print("Result: You are diagnosed with Flu")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Avinash> python -u "c:\Users\Avinash\Desktop\LAB2.2.py"
Do yo have a fever (Y/N): Y
Do yo have a rash (Y/N): Y
Do yo have a cough (Y/N): N
Do yo have a headAche (Y/N): Y
Do yo have a bodyAche (Y/N): N
Do yo have a sneezing (Y/N): N
Do yo have a chills (Y/N): N
Do yo have a runnyNose (Y/N): N
Do yo have a sourThroat (Y/N): Y
Do yo have a swollenGlands (Y/N): N
Do yo have a sneezing (Y/N): Y
Do yo have a chills (Y/N): Y
Do yo have a runnyNose (Y/N): Y
Do yo have a sourThroat (Y/N): Y
Do yo have a swollenGlands (Y/N): N
Do yo have a conjuctives (Y/N): N
Result: You are diagnosed with Measles

Python 3.10.2 64-bit 0 0 0 Ln 11, Col 56 Spaces: 4 UTF-8 CRLF
```

```
LAB2.2.py X
C:\Users\Avinash\Desktop> LAB2.2.py
1 fever = input("Do yo have a fever (Y/N): ")
2 rashes = input("Do yo have a rash (Y/N): ")
3 cough = input("Do yo have a cough (Y/N): ")
4 headAche = input("Do yo have a headAche (Y/N): ")
5 bodyAche = input("Do yo have a bodyAche (Y/N): ")
6 sneezing = input("Do yo have a sneezing (Y/N): ")
7 chills = input("Do yo have a chills (Y/N): ")
8 runnyNose = input("Do yo have a runnyNose (Y/N): ")
9 soreThroat = input("Do yo have a sourThroat (Y/N): ")
10 swollenGlands = input("Do yo have a swollenGlands (Y/N): ")
11 conjunctives = input("Do yo have a conjunctives (Y/N): ")
12 if fever == 'Y' and headAche == 'Y' and runnyNose == 'Y' and rashes == 'Y':
13     print("Result: You are diagnosed with German measles")
14 elif cough == 'Y' and sneezing == 'Y' and runnyNose == 'Y':
15     print("Result: You are diagnosed with Measles")
16 elif fever == 'Y' and rashes == 'Y' and bodyAche == 'Y' and chills == 'Y':
17     print("Result: You are diagnosed with Chicken pox")
18 elif fever == 'Y' and swollenGlands == 'Y':
19     print("Result: You are diagnosed with Mumps")
20 elif fever == 'Y' and headAche == 'Y' and bodyAche == 'Y' and conjunctives == 'Y' and chills == 'Y' and cough == 'Y' and runnyNose == 'Y' and soreThroat == 'Y':
21     print("Result: You are diagnosed with Flu")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Do yo have a runnyNose (Y/N): Y
Do yo have a sourThroat (Y/N): Y
Do yo have a swollenGlands (Y/N): N
Do yo have a conjunctives (Y/N): N
Result: You are diagnosed with Measles
PS C:\Users\Avinash> python -u "c:\Users\Avinash\Desktop\LAB2.2.py"
Do yo have a fever (Y/N): Y
Do yo have a rash (Y/N): N
Do yo have a cough (Y/N): Y
Do yo have a headAche (Y/N): Y
Do yo have a bodyAche (Y/N): Y
Do yo have a sneezing (Y/N): N
Do yo have a chills (Y/N): Y
Do yo have a runnyNose (Y/N): Y
Do yo have a sourThroat (Y/N): Y
Do yo have a swollenGlands (Y/N): N
Do yo have a conjunctives (Y/N): Y
Result: You are diagnosed with Flu
PS C:\Users\Avinash> ]
python 3.10.2 64-bit 0 0 0 Ln 21, Col 48 Spaces: 4 UTF-8 CRLF
```

Result:

Successfully implemented the code to diagnosis a disease for a patient with his given inputs for fever, headAche and others as mentioned.

Avinash reddy Vasipalli

