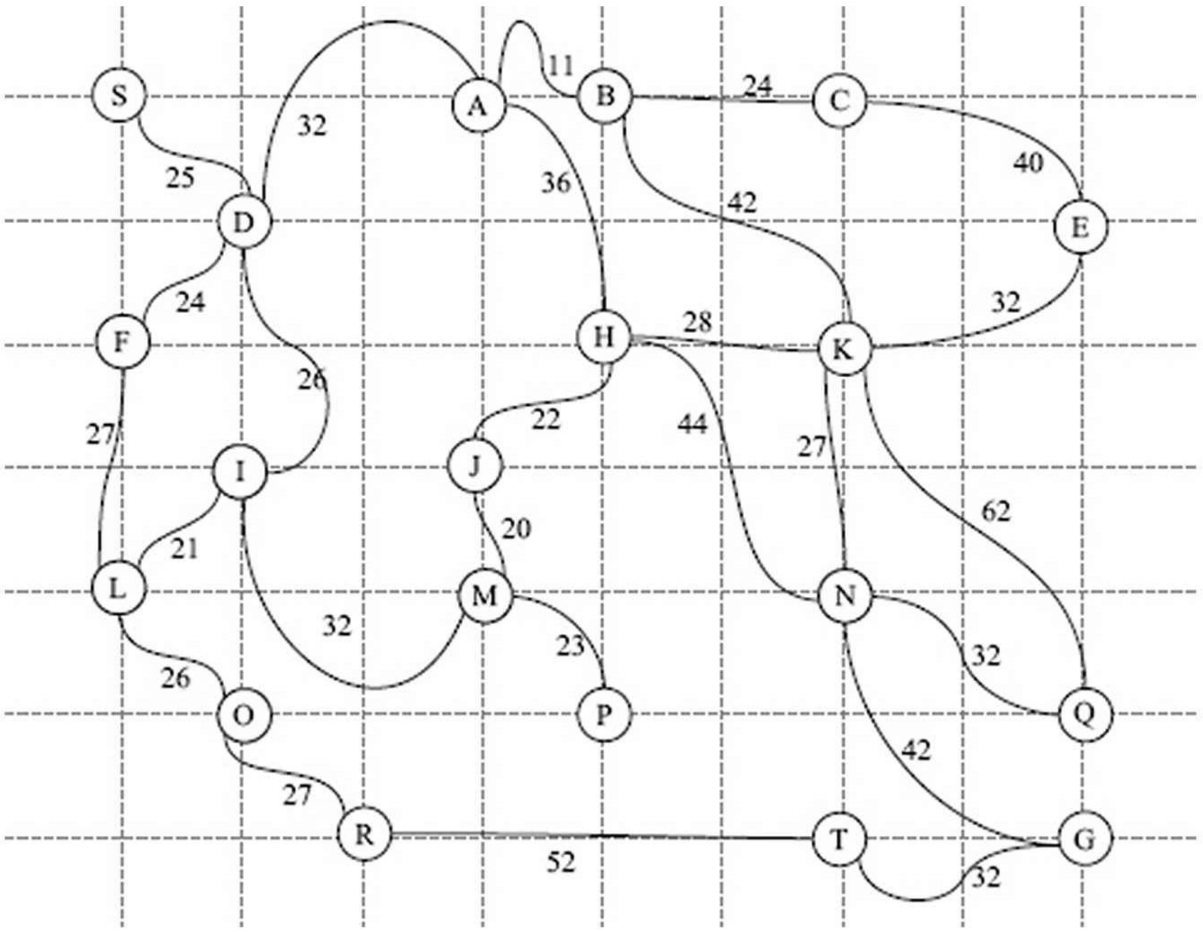
TY CSE AY-2022-23 Sem-I

Artificial Intelligence and Machine Learning Lab

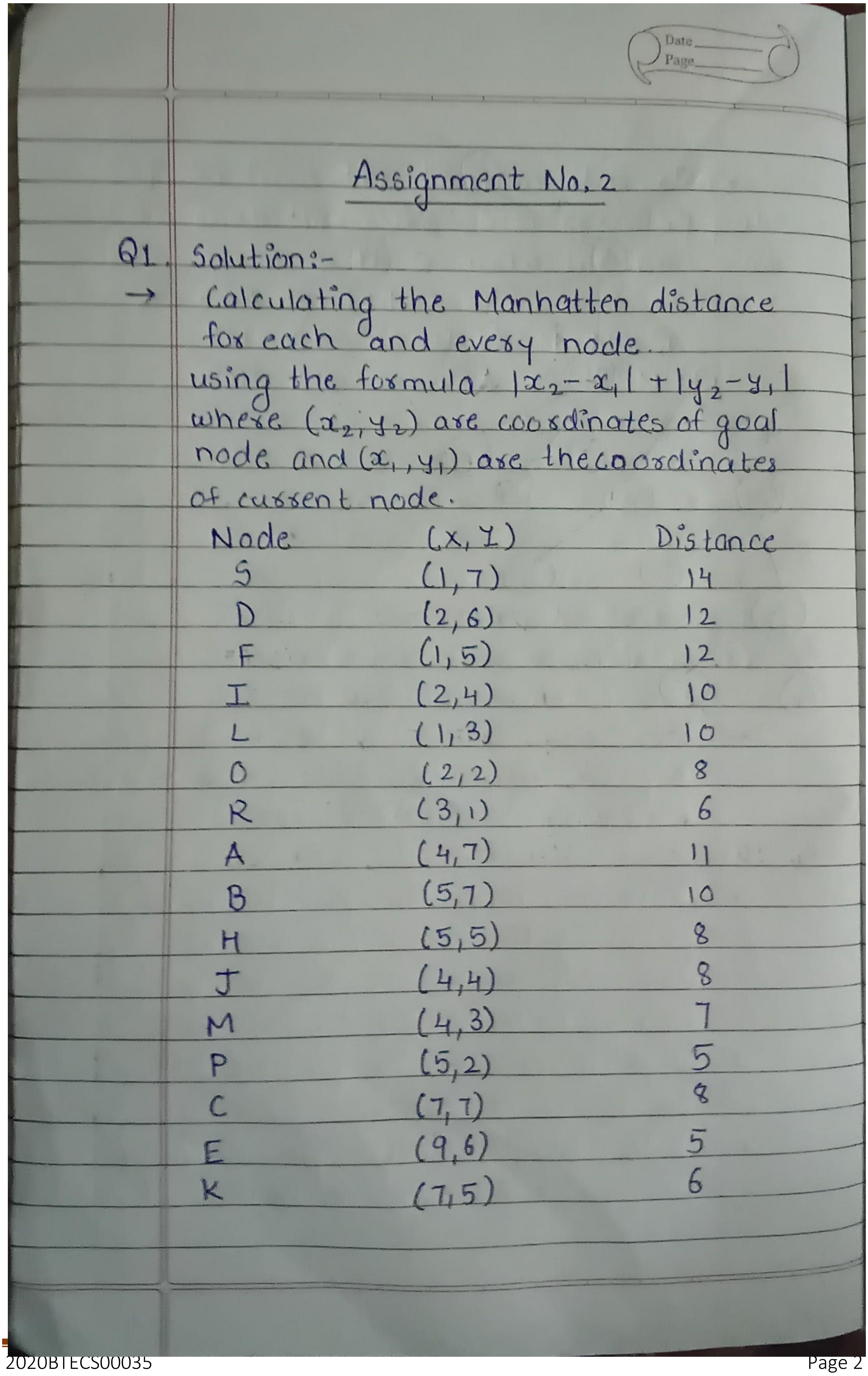
Assignment No 2

Due date- 06/09/2022

1. Consider the below graph, Find the optimal path from start node S to goal node G using A* algorithm.



The length(cost) of each edge is marked on the graph. Use the Manhattan distance as a heuristic function. Assume that each square side on the grid is 10KM.



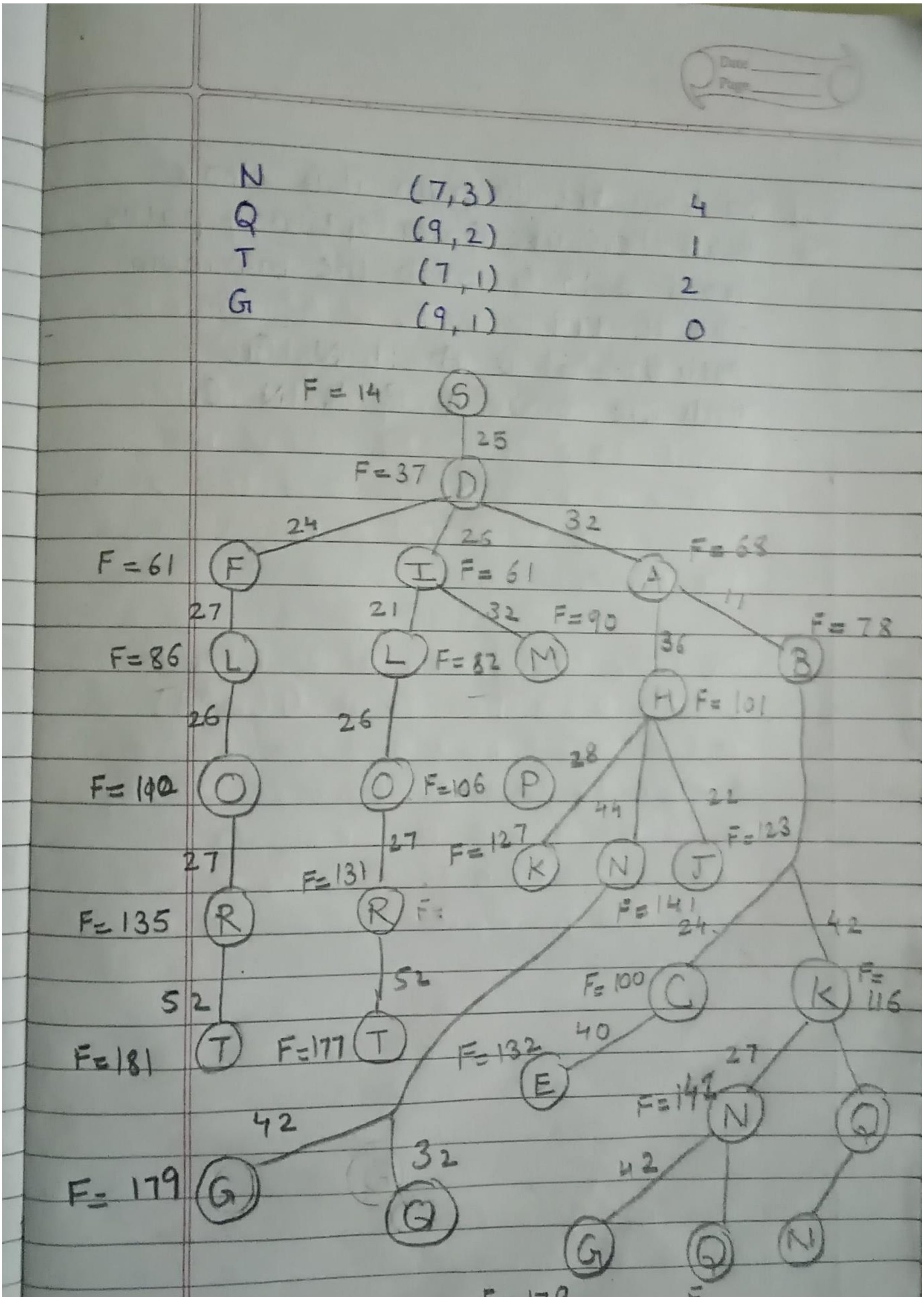
Date
Page

Assignment No. 2

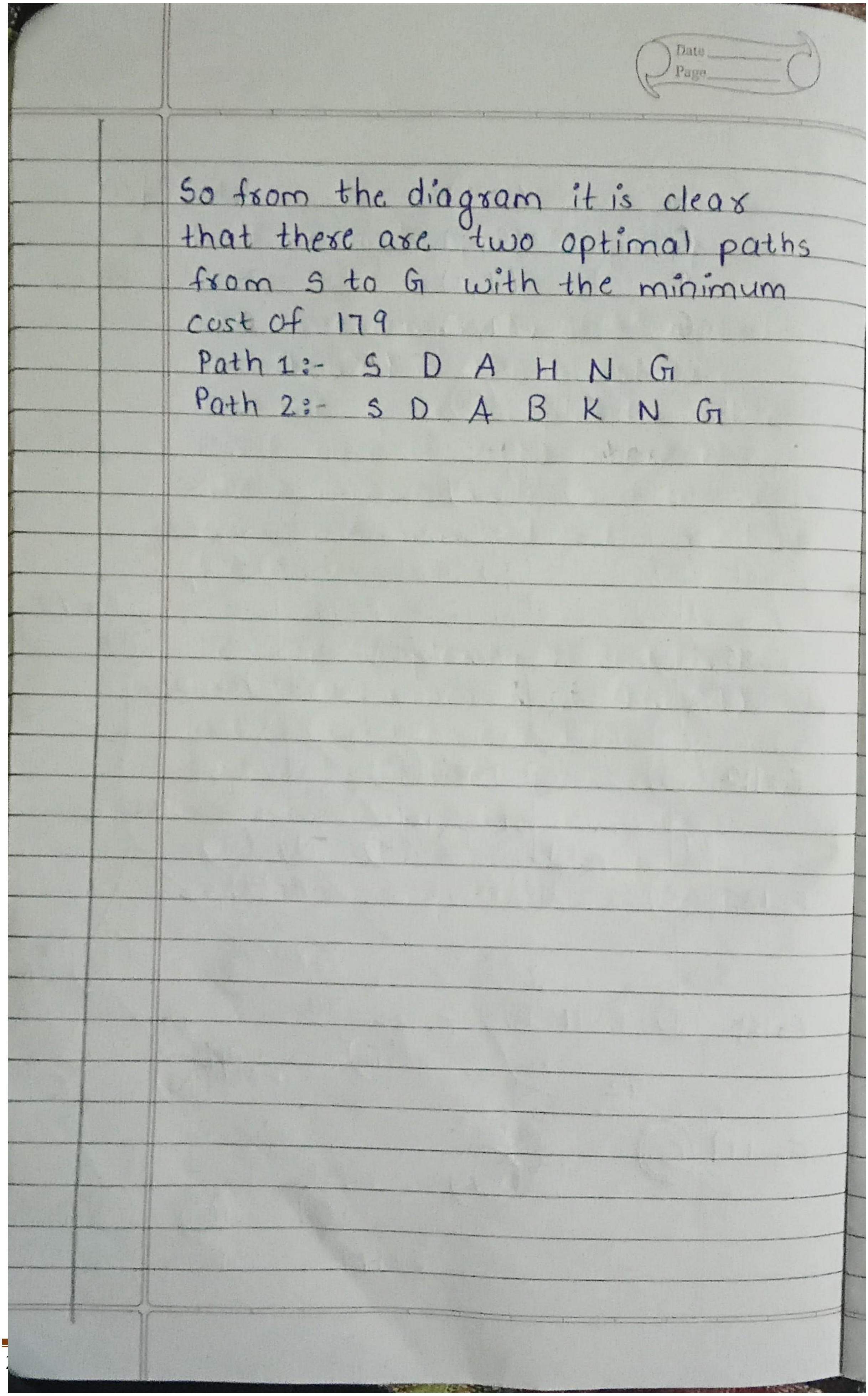
Q1. Solution:-

→ Calculating the Manhatten distance for each and every node.
using the formula $|x_2 - x_1| + |y_2 - y_1|$
where (x_2, y_2) are coordinates of goal node and (x_1, y_1) are thecoordinates of current node.

Node	(x, Y)	Distance
S	(1,7)	14
D	(2,6)	12
F	(1,5)	12
I	(2,4)	10
L	(1,3)	10
O	(2,2)	8
R	(3,1)	6
A	(4,7)	11
B	(5,7)	10
H	(5,5)	8
J	(4,4)	8
M	(4,3)	7
P	(5,2)	5
C	(7,7)	8
E	(9,6)	5
K	(7,5)	6



N	(7,3)	4
Q	(9,2)	1
T	(7,1)	2
G	(9,1)	0

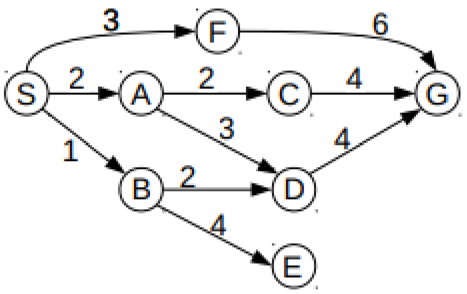


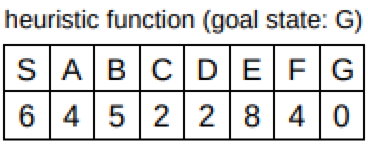
Date
Page

So from the diagram it is clear that there are two optimal paths from S to G with the minimum cost of 179

Path 1:- S D A H N G
Path 2:- S D A B K N G

2. Consider below a directed graph given and corresponding heuristic function values given in the table.

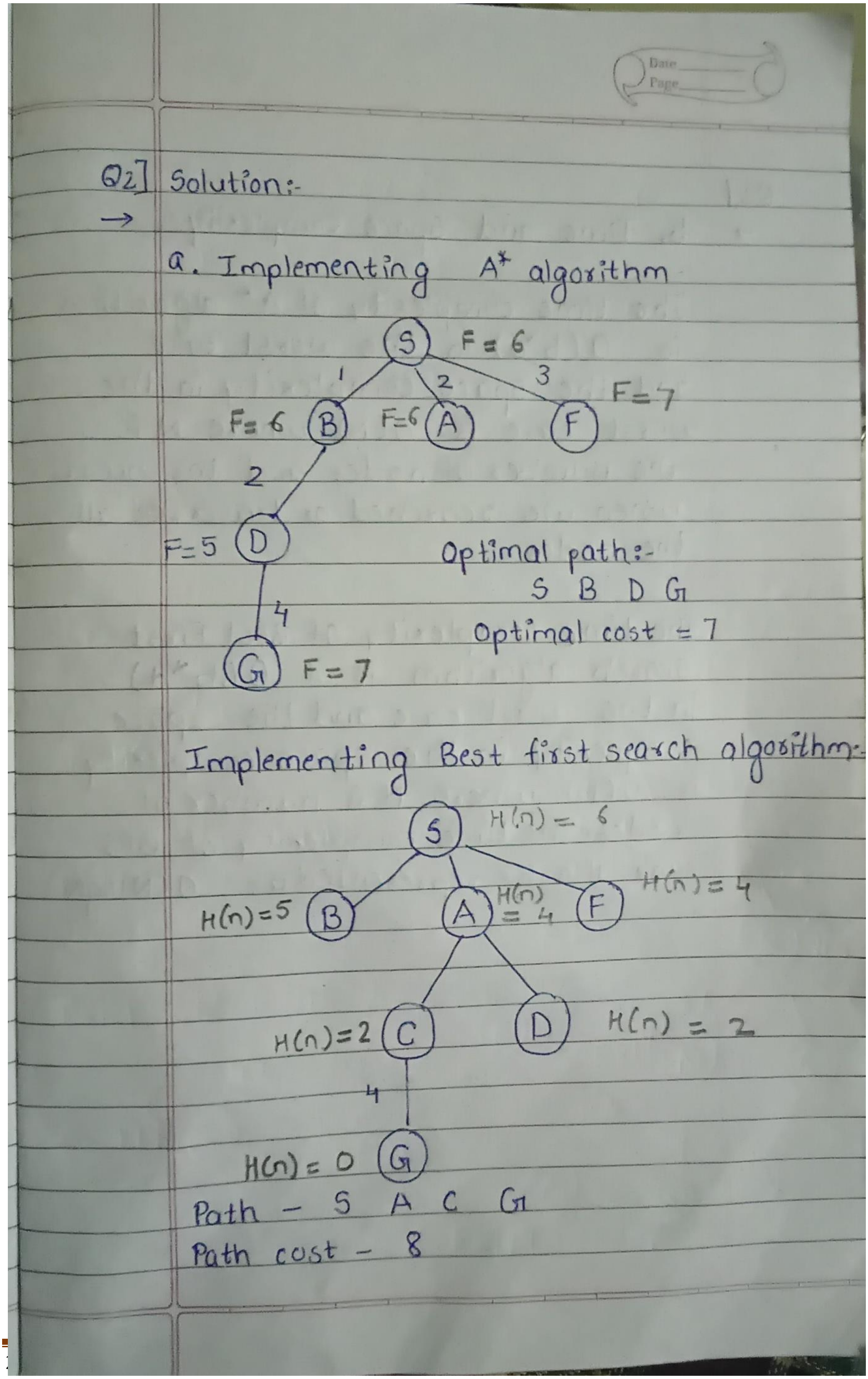



heuristic function (goal state: G)

S	A	B	C	D	E	F	G
6	4	5	2	2	8	4	0

a. Implement A* algorithm and Best first search algorithm to identify an optimal path from Starting state S to goal state G.
b. What will be the time and space complexity?

2020BTECS00035
Vyankatesh Sanjay Bura



Date
Page

Q2] Solution:-

→

a. Implementing A* algorithm

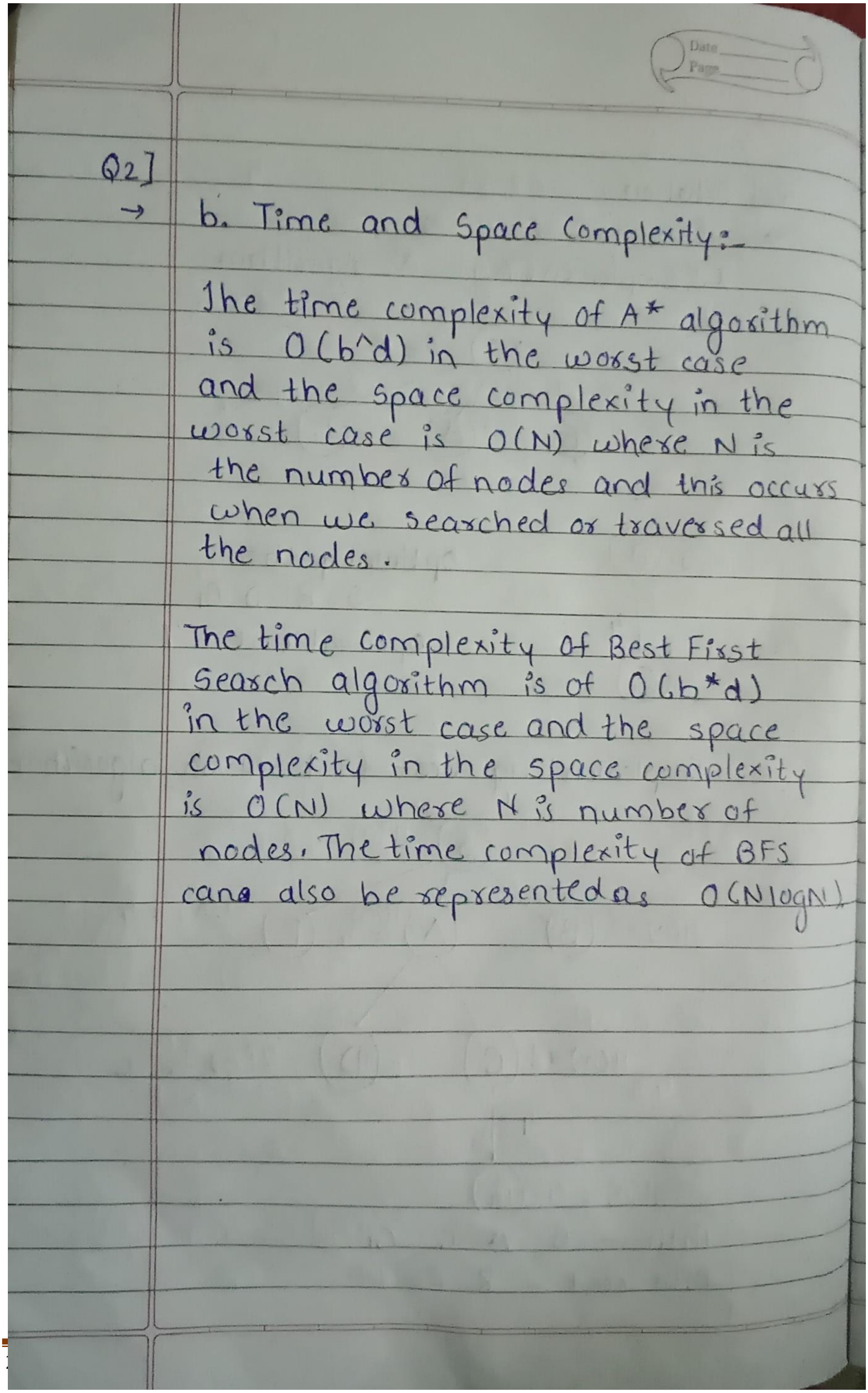
S F = 6
1
2
3
F = 7
F = 6 B
F=6 A
F
2
F=5 D
4
G F = 7

Optimal path:-
S B D G
Optimal cost = 7

Implementing Best first search algorithm:-

S H(n) = 6
H(n)=5 B
A H(n) = 4
F H(n) = 4
H(n)=2 C
D H(n) = 2
4
H(n) = 0 G

Path - S A C G
Path cost - 8



Date
Page

Q2]

→ b. Time and Space Complexity:-

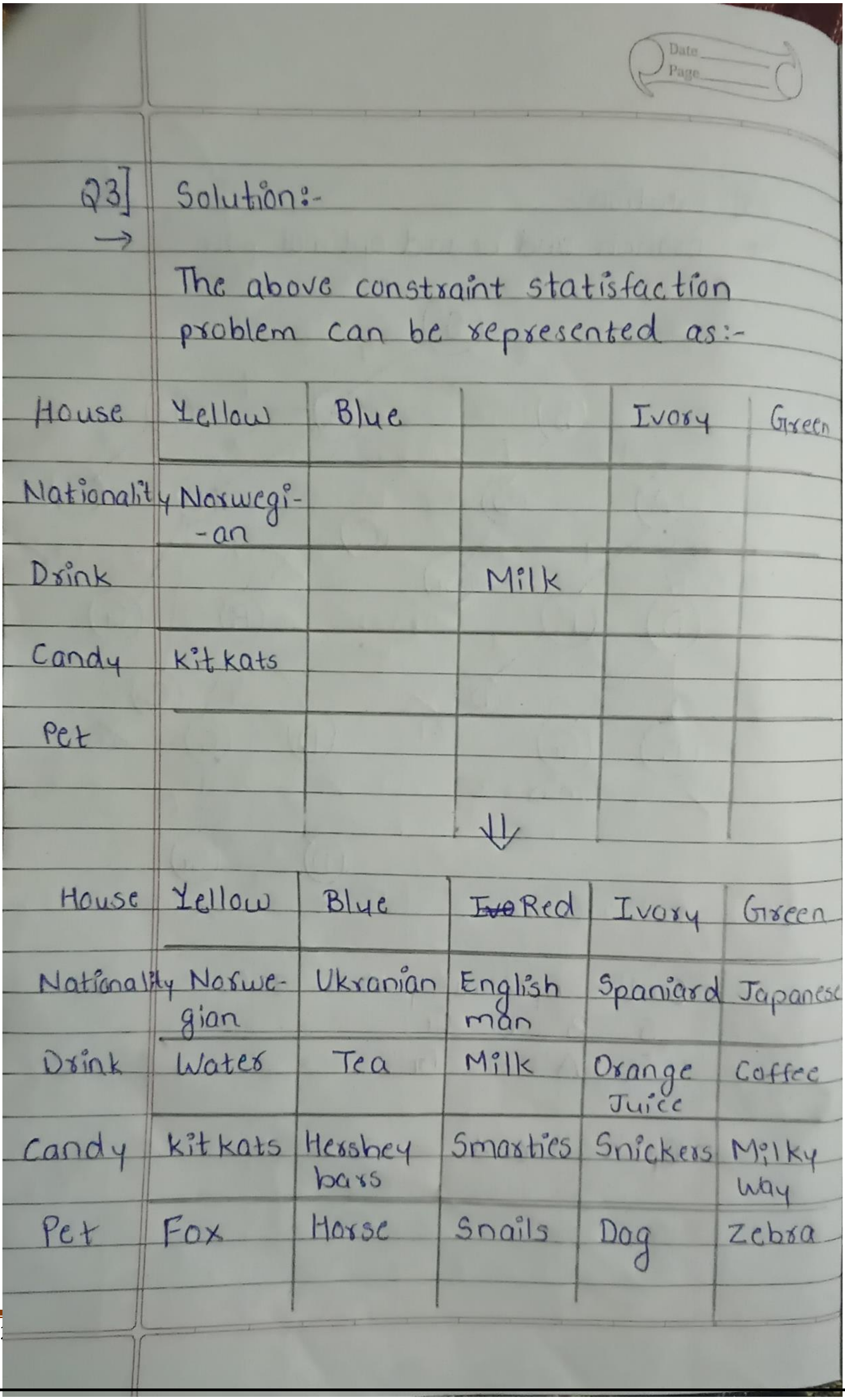
Jhe time complexity of A* algorithm is O(b^d) in the worst case and the space complexity in the worst case is O(N) where N is the number of nodes and this occurs when we searched or traversed all the nodes.

The time complexity of Best First Search algorithm is of O(b*d) in the worst case and the space complexity in the space complexity is O(N) where N is number of nodes. The time complexity of BFS cana also be representedas O(NlogN)

3. Consider the following logic puzzle: In five houses, each with a different color, live five persons of different nationalities, each of whom prefers a different brand of candy, a different drink, and a different pet. Given the following facts, the questions to answer are “Where does the zebra live, and in which house do they drink water?”

The Englishman lives in the red house.
The Spaniard owns the dog.
The Norwegian lives in the first house on the left.
The green house is immediately to the right of the ivory house.
The man who eats Hershey bars lives in the house next to the man with the fox.
Kit Kats are eaten in the yellow house.
The Norwegian lives next to the blue house.
The Smarties eater owns snails.
The Snickers eater drinks orange juice.
The Ukrainian drinks tea.
The Japanese eats Milky Ways.
Kit Kats are eaten in a house next to the house where the horse is kept.
Coffee is drunk in the green house.
Milk is drunk in the middle house.

Discuss different representations of this problem as a CSP. Why would one prefer one representation over another?



Date
Page

Q3] Solution:-
→

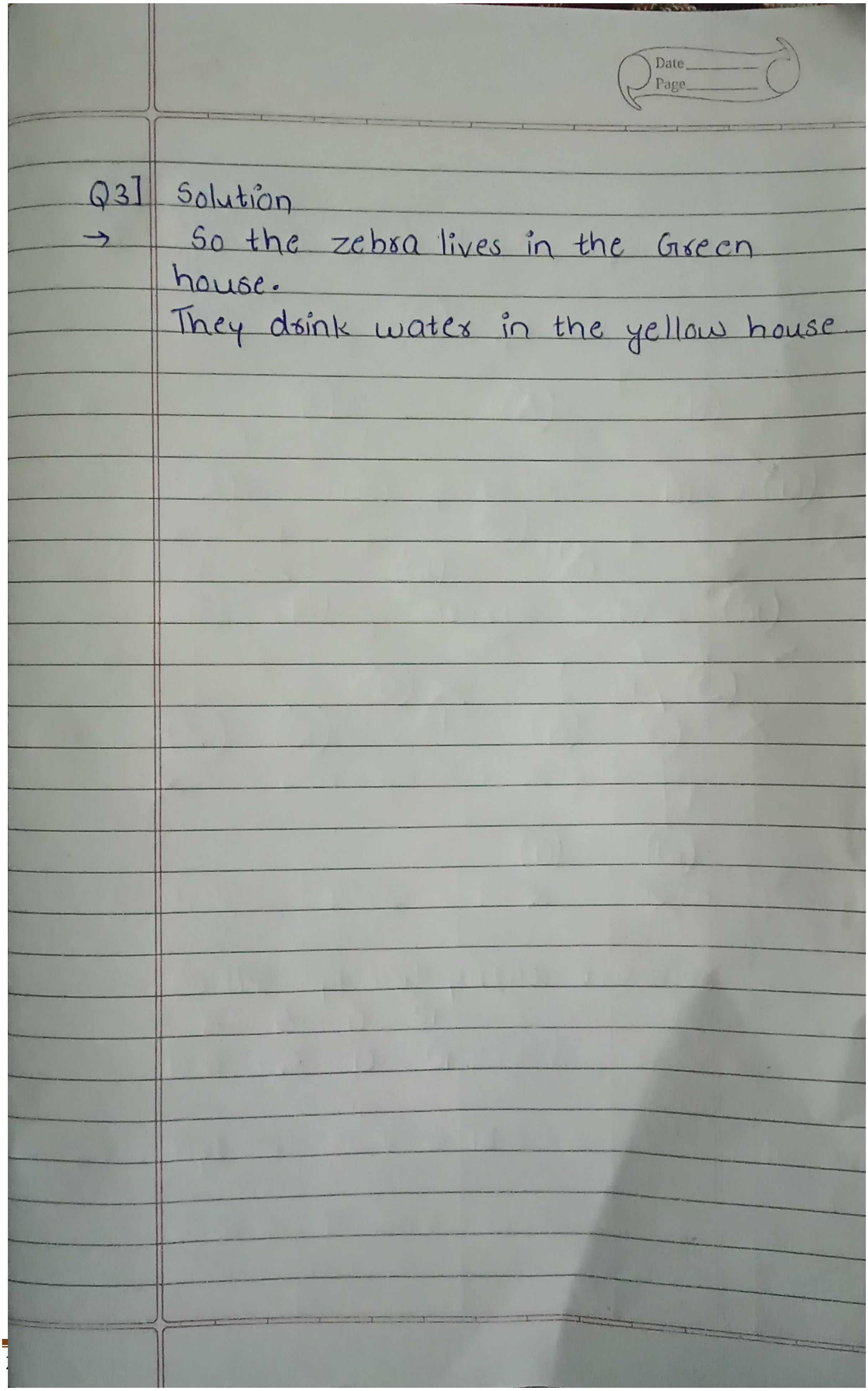
The above constraint statisfaction problem can be represented as:-

House	Yellow	Blue		Ivory	Green
Nationality	Norwegi- -an				
Drink			Milk		
Candy	kitkats				
Pet					

⇓

House	Yellow	Blue	~~Ivo~~Red	Ivory	Green
Nationality	Norwegian	Ukranian	Englishman	Spaniard	Japanese
Drink	Water	Tea	Milk	Orange Juice	Coffee
Candy	kitkats	Hershey bars	Smarties	Snickers	Milky way
Pet	Fox	Horse	Snails	Dog	Zebra

2020BTECS00035
Vyankatesh Sanjay Bura



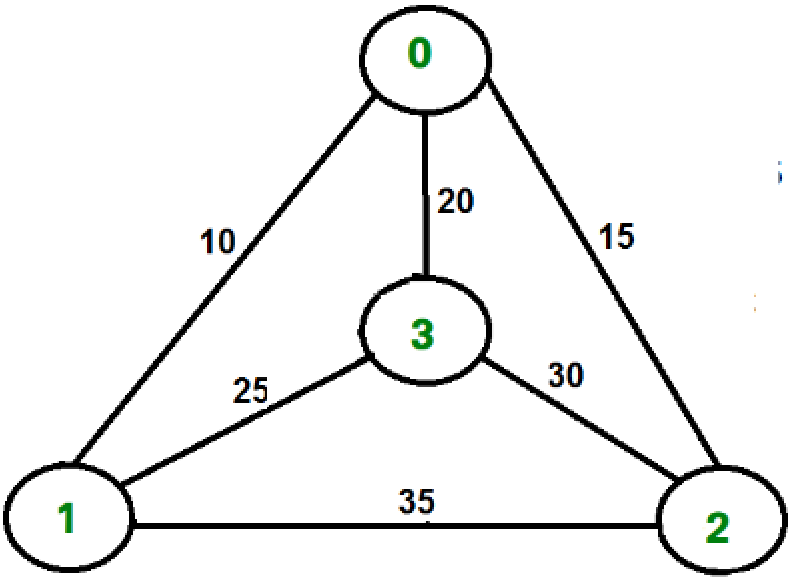
Date
Page

Q3] Solution

→ So the zebra lives in the Green house.

They drink water in the yellow house

4. Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible tour that visits every city exactly once and returns to the starting point.

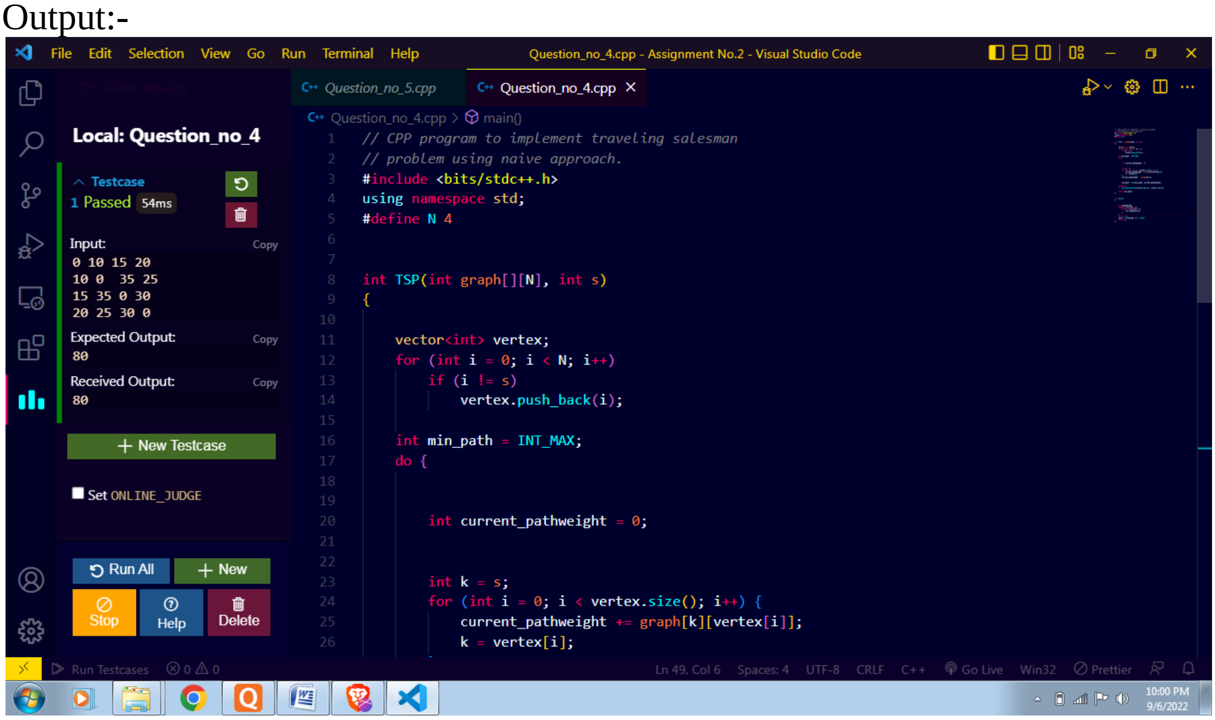


Identify the optimal path and implement this TS problem using branch and bound concept.

Code:-

```
#include <bits/stdc++.h>
using namespace std;
#define N 4


int TSP(int graph[][N], int s)
{

    vector<int> vertex;
    for (int i = 0; i < N; i++)
        if (i != s)
            vertex.push_back(i);

    int min_path = INT_MAX;
    do {


        int current_pathweight = 0;
        int k = s;
        for (int i = 0; i < vertex.size(); i++) {
            current_pathweight += graph[k][vertex[i]];
            k = vertex[i];
        }
        current_pathweight += graph[k][s];
        min_path = min(min_path, current_pathweight);

    } while (
        next_permutation(vertex.begin(), vertex.end()));
    return min_path;
}


int main()
{
    int graph[N][N];
    for(int i=0;i<N;i++){
        for(int j=0;j<N;j++){
            cin>>graph[i][j];
        }
    }
    int s = 0;
    cout << TSP(graph, s) << endl;
    return 0;
}
```

Output:-



Input:

```
0 10 15 20
10 0  35 25
15 35 0 30
20 25 30 0
```

Expected Output:

```
80
```

Received Output:

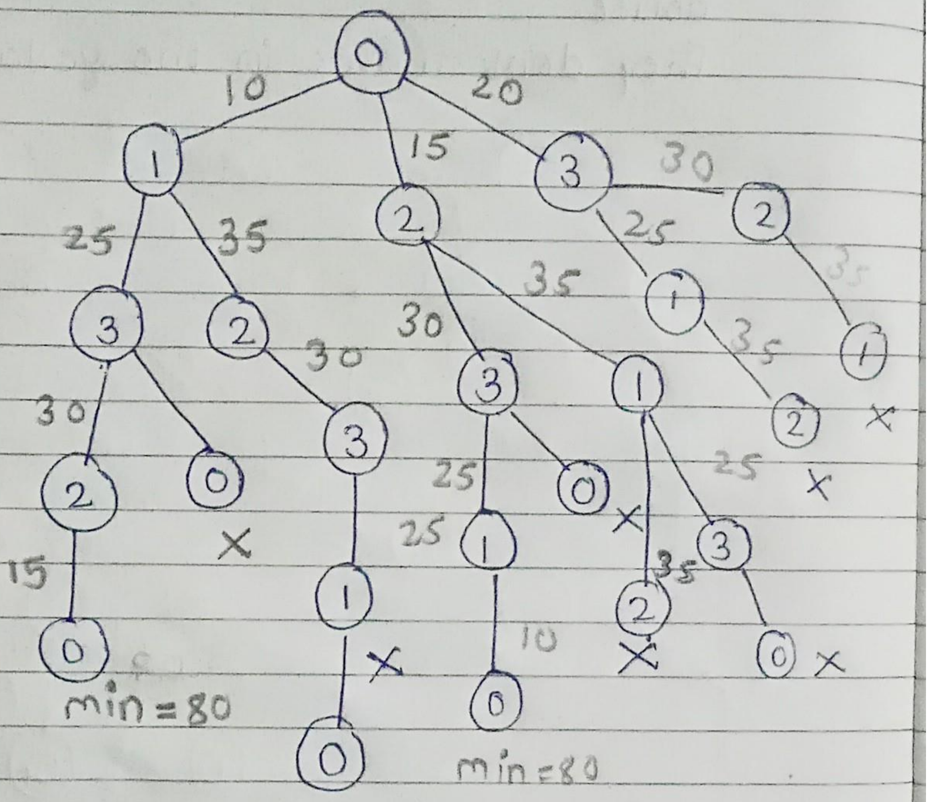
```
80
```

□ Set ONLINE_JUDGE

```
// CPP program to implement traveling salesman
// problem using naive approach.
#include <bits/stdc++.h>
using namespace std;
#define N 4


int TSP(int graph[][N], int s)
{

    vector<int> vertex;
    for (int i = 0; i < N; i++)
        if (i != s)
            vertex.push_back(i);

    int min_path = INT_MAX;
    do {


        int current_pathweight = 0;


        int k = s;
        for (int i = 0; i < vertex.size(); i++) {
            current_pathweight += graph[k][vertex[i]];
            k = vertex[i];
```

Q4]

→ Solution:-



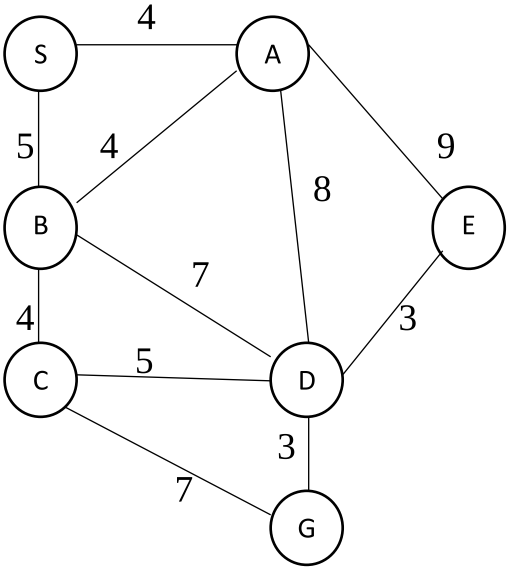
So the optimal path by branch and bound is

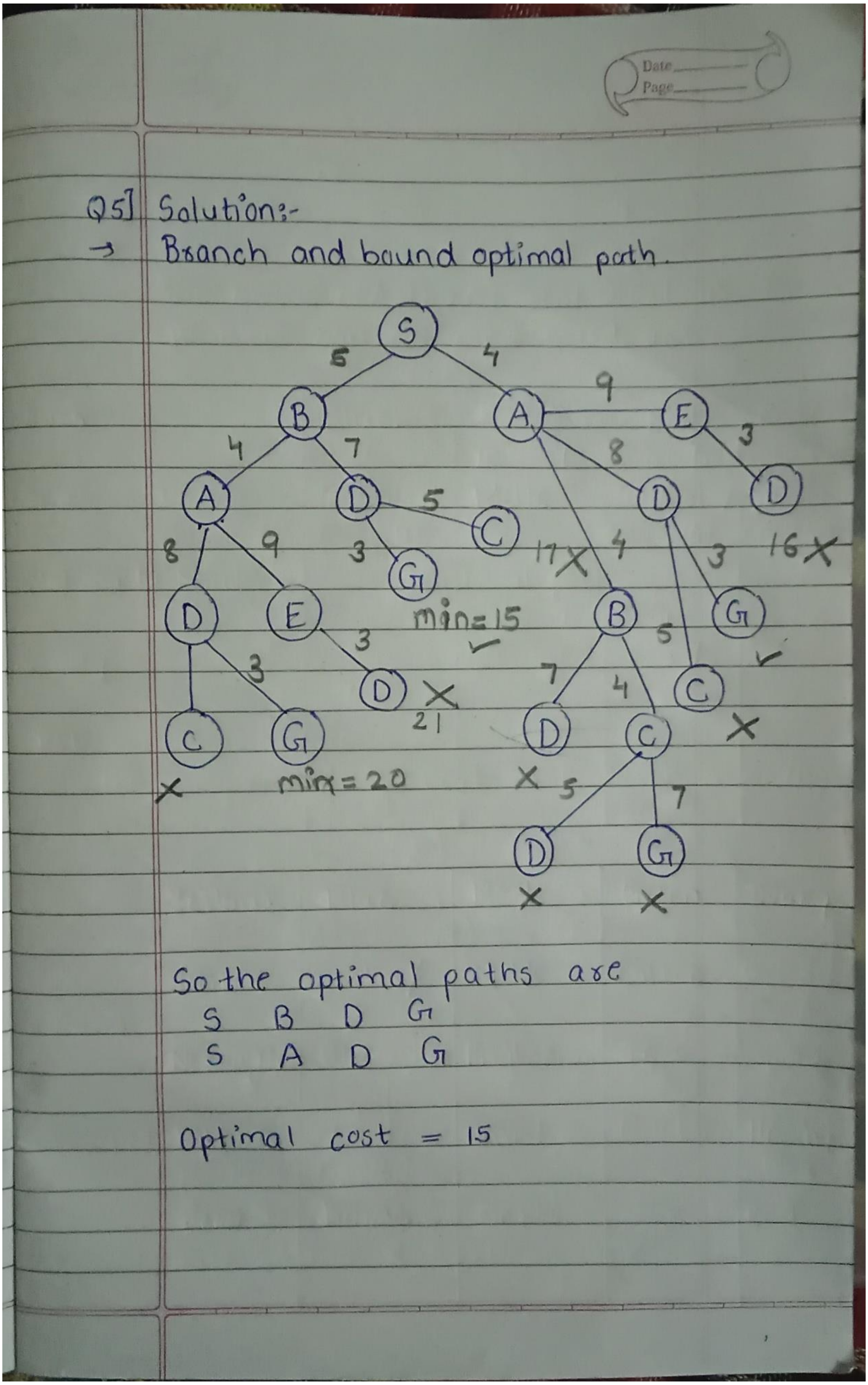
→ 0 1 3 2 0

→ 0 2 3 1 0

Minimum cost = 80

5. Using branch and bound identify optimal path of below graph from starting state S to goal state G.





Q5] Solution:-

→ Branch and bound optimal path.

S
5
4
B
A
9
E
4
7
8
3
A
D
D
D
5
C
8
9
3
17 ✗
4
3
16 ✗
G
D
E
min=15 ✓
B
G
3
5
✓
3
D ✗
7
4
C
21
✗
C
G
D
C
✗
min=20
✗
5
7
D
G
✗
✗

So the optimal paths are
S B D G
S A D G

Optimal cost = 15

****