GHARDA FOUNDATION
GHARDA INSTITUTE OF TECHNOLOGY, LAVEL
Department of Computer Engineering

Evaluation Sheet

Class: TE-Computer Engineering Sem: V

Subject: **Artificial Intelligence Lab(CSL604)**

Experiment No: 5

Title of Experiment: Study the implementation of Greedy Best First Search Algorithm.

Name of Student: Niraj Nitin Surve Roll No: 68

Date of Performance:

Sr. No.	Evaluation Criteria	Max Marks	Marks Obtained
1	Practical Performance	8	
2	Oral	5	
3	Timely Submission	2	
	Total	15	

Signature of Subject Teacher
(Prof. M. A. Khandke)

Program Code –

```
from queue import PriorityQueue


def greedy_best_first_search(graph, start, goal, weights):
    frontier = PriorityQueue()
    frontier.put(start, 0)
    came_from = {}
    came_from[start] = None

    while not frontier.empty():
        current = frontier.get()

        if current == goal:
            break

        for neighbor in graph[current]:
            if neighbor not in came_from:
                priority = weights[neighbor]
                frontier.put(neighbor, priority)
                came_from[neighbor] = current
    return came_from
graph = {}
n = int(input("Enter the number of nodes: "))

for i in range(n):
    node = input("Enter the node: ")
    neighbors = input("Enter the neighbors separated by spaces: ")
    graph[node] = neighbors.split()

weights = {}
for node in graph:
```

```
    weight = int(input("Enter the weight of node {}: ".format(node)))
    weights[node] = weight

start = input("Enter the start node: ")
goal = input("Enter the goal node: ")

came_from = greedy_best_first_search(graph, start, goal, weights)

path = []
current = goal
while current != start:
    path.append(current)
    current = came_from[current]
path.append(start)
path.reverse()

print("Shortest path:", path)
```

Output –

```
Enter the number of nodes: 6
Enter the node: P
Enter the neighbors separated by spaces: Q R S
Enter the node: Q
Enter the neighbors separated by spaces: P X
Enter the node: R
Enter the neighbors separated by spaces: P X Y
Enter the node: S
Enter the neighbors separated by spaces: P
Enter the node: X
Enter the neighbors separated by spaces: Q R Y
Enter the node: Y
Enter the neighbors separated by spaces: R X
Enter the weight of node P: 6
Enter the weight of node Q: 20
Enter the weight of node R: 4
Enter the weight of node S: 11
Enter the weight of node X: 9
Enter the weight of node Y: 3
Enter the start node: P
Enter the goal node: Y
Shortest path: ['P', 'R', 'Y']
```