

**OUTPUT:**

```
PS C:\Users\dell\Abinash Regmi> python
Adjacency List:
A -> ['B', 'C']
B -> ['D', 'E']
C -> ['F']
D -> []
E -> []
F -> []

Enter the start node: a
Enter the destination node: f

Path from A to F:
A -> B -> E -> C -> F
```

**OUTPUT:**

```
PS C:\Users\dell\Abinash Regmi> python -u "c:\u
i\Semester-IV\AI\lab5.py"
Graph adjacency list with costs:
A -> [('B', 1), ('C', 4), ('D', 3)]
B -> [('E', 2)]
C -> [('E', 1), ('F', 5)]
D -> [('F', 6)]
E -> [('F', 1)]
F -> []

Searching for shortest path from 'A' to 'F'...

Shortest path found: A -> C -> E -> F
Total cost: 6
```

**OUTPUT:**

```
PS C:\Users\dell\Abinash Regmi> python -u 'C:\Users\dell\Abinash Regmi\Semester-IV\AI\lab6.py'
The optimal value is: 3
```

**OUTPUT:**

```
PS C:\Users\dell\Abinash Regmi> python -u "c:\Users\dell\Abinash Regmi\Semester-IV\AI\lab7.py"
Solution found! Here is the step-by-step path:
Step 0: Jug A = 0 | Jug B = 0
Step 1: Jug A = 0 | Jug B = 3
Step 2: Jug A = 3 | Jug B = 0
Step 3: Jug A = 3 | Jug B = 3
Step 4: Jug A = 5 | Jug B = 1
Step 5: Jug A = 0 | Jug B = 1
Step 6: Jug A = 1 | Jug B = 0
```