02a. A-Star Algorithm

In [1]:
```python
import heapq

def a_star_search(graph, start, goal, heuristic, cost):
    # Priority queue for exploring nodes
    priority_queue = []
    heapq.heappush(priority_queue, (0 + heuristic[start], start))
    visited = set()
    g_cost = {start: 0}
    parent = {start: None}

    while priority_queue:
        current_cost, current_node = heapq.heappop(priority_queue)

        if current_node in visited:
            continue

        visited.add(current_node)

        if current_node == goal:
            break

        for neighbor in graph[current_node]:
            new_cost = g_cost[current_node] + cost[(current_node, neighbor)]
            if neighbor not in g_cost or new_cost < g_cost[neighbor]:
                g_cost[neighbor] = new_cost
                f_cost = new_cost + heuristic[neighbor]
                heapq.heappush(priority_queue, (f_cost, neighbor))
                parent[neighbor] = current_node

    path = []
    node = goal
    while node is not None:
        path.append(node)
        node = parent[node]
    path.reverse()

    return path
```

In [2]:
```python
# Example graph
graph = {
    'A': ['B', 'C'],
    'B': ['D', 'E'],
    'C': ['F', 'G'],
    'D': [],
    'E': [],
    'F': [],
    'G': []
}

# Example heuristic values (assumed for demonstration)
heuristic = {
    'A': 6,
    'B': 4,
    'C': 4,
    'D': 0,
    'E': 2,
    'F': 3,
    'G': 1
}

# Example costs between nodes (assumed for demonstration)
cost = {
    ('A', 'B'): 1,
    ('A', 'C'): 1,
    ('B', 'D'): 1,
    ('B', 'E'): 3,
    ('C', 'F'): 5,
    ('C', 'G'): 2
}

start = 'A'
goal = 'D'

path = a_star_search(graph, start, goal, heuristic, cost)
print("A* Search Path:", path)
```

```
A* Search Path: ['A', 'B', 'D']
```

02b. Contour Plot

In [3]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

dataset = pd.read_csv('./corolla.csv')
x = dataset['KM']
y = dataset['Weight']
z = dataset['Price']

plt.tricontourf(x, y, z, levels=20, cmap='jet')
plt.colorbar(label='Price')
plt.xlabel('KM')
plt.ylabel('Weight')
plt.title('Contour Plot')
plt.show()
```