```
n



f')


ted and dist[v] < min_dist:
ist[v]




= float('inf') and v not in visited:
 graph[u][v] < dist[v]:
= dist[u] + graph[u][v]




 float('inf')],
float('inf')],
2],
f'), float('inf'), 0, 7],
f'), float('inf'), 9, 0]
```

```
[0, 7, 3, 9, 5]

=== Code Execution Successful ===
```

```
rce, target, n):
 in range(n)}
:
(v, w))
(u, w))


] * n



ppop(pq)


target]



h[u]:
 w < dist[v]:
 = dist[u] + w
eappush(pq, (dist[v], v))



, (0, 5, 14),
5),
),



urce, target, n))
```

```
[0, 7, 3, 9, 5]

=== Code Execution Successful ===
```

```
freq):


her.freq

aracters, frequencies):


ap, Node(characters[i], frequencies[i]))


min_heap)
(min_heap)
 left.freq + right.freq)

t
ap, new_node)

t):

de.char, code))
```

```
[0, 7, 3, 9, 5]

=== Code Execution Successful ===
```

```
]


]
ge(n):
appush(min_heap, Node(characters[i], frequencies[i]))

n_heap) > 1:
eapq.heappop(min_heap)
heapq.heappop(min_heap)
 = Node(None, left.freq + right.freq)
.left = left
.right = right
appush(min_heap, new_node)

, code, result):
char:
lt.append((node.char, code))

ode.left:
dfs(node.left, code + "0", result)
ode.right:
dfs(node.right, code + "1", result)

.heappop(min_heap)
s = []
, huffman_codes)

an_codes


', 'b', 'c', 'd']
, 9, 12, 13]
uffman_codes(characters, frequencies))
```

```
[0, 7, 3, 9, 5]

=== Code Execution Successful ===
```

ain.py

```
    d = [sys.maxsize] * n
    d[s] = 0

    pq = [(0, s)]

    while pq:
        curr_d, u = heapq.heappop(pq)

        if curr_d > d[u]:
            continue

        for v in range(n):
            w = g[u][v]
            if w < sys.maxsize:
                dist = curr_d + w
                if dist < d[v]:
                    d[v] = dist
                    heapq.heappush(pq, (dist, v))

    return d

n1 = 5
g1 = [
    [0, 10, 3, float('inf'), float('inf')],
    [float('inf'), 0, 1, 2, float('inf')],
    [float('inf'), 4, 0, 8, 2],
    [float('inf'), float('inf'), float('inf'), 0, 7],
    [float('inf'), float('inf'), float('inf'), 9, 0]
]
s1 = 0
print(dijkstra_adj_matrix(n1, g1, s1))

```

input

```
, 7, 3, 9, 5]


.Program finished with exit code 0
ess ENTER to exit console.
```