Shortest Path in Weighted undirected graph

You are given a weighted undirected graph having **n+1** vertices numbered from 0 to n and **m** edges describing there are edges between a to b with some weight, find the shortest path between the vertex 1 and the vertex n and if path does not exist then return a list consisting of only -1.

```
Example:
Input:
n = 5, m= 6
edges = [[1,2,2], [2,5,5], [2,3,4],
[1,4,1],[4,3,3],[3,5,1]]
Output:
1 4 3 5
Explaination:
Shortest path from 1 to n is by the path 1 4 3 5
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **shortestPath()** which takes n vertex and m edges and vector of edges having weight as inputs and returns the shortest path between vertex 1 to n.

Expected Time Complexity: O(m* log(n))
Expected Space Complexity: O(n)

Constraint:
$2 <= n <= 10^5$
$0 <= m <= 10^5$
0<= a, b <= n
$1 <= w <= 10^5$