**Count the paths:**

Given a directed acyclic graph(DAG) with n nodes labeled from 0 to n-1. Given edges, s and d ,count the number of ways to reach from s to d.There is a directed Edge from vertex edges[i][0] to the vertex edges[i][1].

**Example:**

**Input:** edges = {{0,1},{0,3},{1,2},{3,2}},

n = 4, s = 0, d = 2

**Output:** 2

**Explanation:** There are two ways to reach at

2 from 0. These are-

1. 0->1->2

2. 0->3->2

**Your Task:**  
You don't need to read or print anything. Your task is to complete the function **possible\_paths()**which takes edges, n, s and d as input parameter and returns the number of ways to reach from s to d.

**Expected Time Compelxity:**O(2n)  
**Expected Space Complexity:**O(n+e)

where e is the number of edges in the graph.