**Problem Name:** Reconstruct Itineary

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

Doge is solving a DSA problem but he is having some difficulty with it. Can you help Doge so that he can complete it and play with his friend Cheem?

Problem is:

**Input Format:**

The first line of input is an integer value n (total no of vertex in the graph).

The Second line of input contains integer value m (total no of edges in the graph)

The next m lines contain two space-separated integers forming a graph.

Last line of input contains the integer value source and destination

**Output Format:**

**Constraints:**

**Examples:**

**Approach one Solution:**

**Explanation:**

algorithm is often called Fleury's algorithm. But actually it is Hierholzer's algorithm according to the wiki. Anyway, it works like this:

Keep going one path until stuck, then retreat and push the vertices along the route to a stack until it reaches a vertex that has alternative paths, then go along that path and repeat the process.  
The assumption for this to work is there is guaranteed to exist one Euler path. (This problem is basically to find a Euler path of a graph).

**Code:**

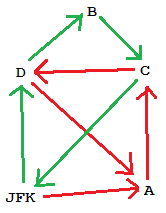
**Time Complexity**:

**Space Complexity:**

**Approach second Solution:**

Explanation: First keep going forward until you get stuck. That's a good main path already. Remaining tickets form cycles which are found on the way back and get merged into that main path. By writing down the path backwards when retreating from recursion, merging the cycles into the main path is easy - the end part of the path has already been written, the start part of the path hasn't been written yet, so just write down the cycle now and then keep backwards-writing the path.

Example:



From JFK we first visit JFK -> A -> C -> D -> A. There we're stuck, so we write down A as the end of the route and retreat back to D. There we see the unused ticket to B and follow it: D -> B -> C -> JFK -> D. Then we're stuck again, retreat and write down the airports while doing so: Write down D before the already written A, then JFK before the D, etc. When we're back from our cycle at D, the written route is D -> B -> C -> JFK -> D -> A. Then we retreat further along the original path, prepending C, A and finally JFK to the route, ending up with the route JFK -> A -> C -> D -> B -> C -> JFK -> D -> A.

**Code:**

**Time Complexity: O(ElogE)**

**Space Complexity:** O(E)