**Reconstruct Itinerary:**

Given a list of airline tickets represented by pairs of departure and arrival airports [from, to], reconstruct the itinerary in order. All of the tickets belong to a man who departs from JFK. Thus, the itinerary must begin with JFK.

**Note:**

1. If there are multiple valid itineraries, you should return the itinerary that has the smallest lexical order when read as a single string. For example, the itinerary ["JFK", "LGA"] has a smaller lexical order than ["JFK", "LGB"].
2. All airports are represented by three capital letters (IATA code).
3. You may assume all tickets form at least one valid itinerary.
4. One must use all the tickets once and only once.

**Example 1:**

**Input:** [["MUC", "LHR"], ["JFK", "MUC"], ["SFO", "SJC"], ["LHR", "SFO"]]

**Output:** ["JFK", "MUC", "LHR", "SFO", "SJC"]

**Example 2:**

**Input:** [["JFK","SFO"],["JFK","ATL"],["SFO","ATL"],["ATL","JFK"],["ATL","SFO"]]

**Output:** ["JFK","ATL","JFK","SFO","ATL","SFO"]

**Explanation:** Another possible reconstruction is ["JFK","SFO","ATL","JFK","ATL","SFO"].

  But it is larger in lexical order.

**Example 3:**

**Input:**

[["JFK","KTL"],["JFK","NTL"],["NTL","JFK"]]

**Output:**

["JFK","NTL","JFK","KTL"]

Here if we follow lexicographical order we cannot traverse all the paths.

JFK->KTL (Not able to reach any further and all paths are not traversed.

So the correct answer will be

JFK->NTL->JFK->KTL