

332. Reconstruct Itinerary

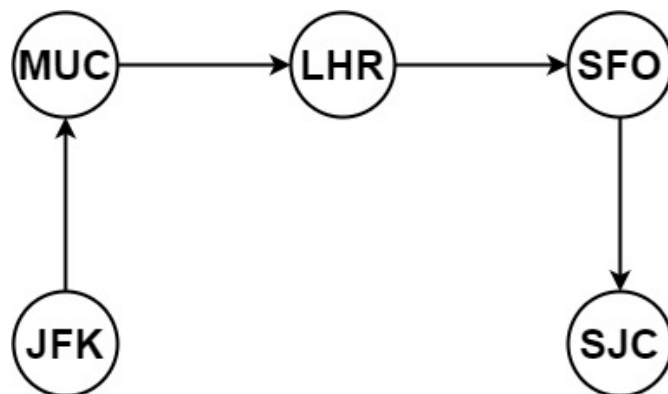
You are given a list of airline tickets where `tickets[i] = [fromi, toi]` represent the departure and the arrival airports of one flight. Reconstruct the itinerary in order and return it.

All of the tickets belong to a man who departs from "JFK", thus, the itinerary must begin with "JFK". 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"]`.

You may assume all tickets form at least one valid itinerary. You must use all the tickets once and only once.

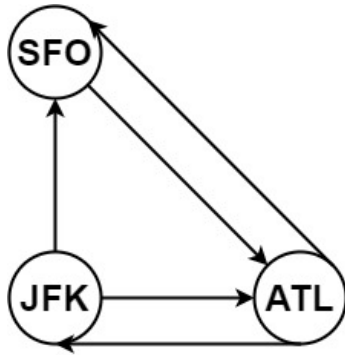
Example 1:



Input: `tickets = [["MUC","LHR"],["JFK","MUC"],["SFO","SJC"],["LHR","SFO"]]`

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

Example 2:



Input: tickets = [["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.

Constraints:

- $1 \leq \text{tickets.length} \leq 300$
- $\text{tickets}[i].\text{length} == 2$
- $\text{fromi.length} == 3$
- $\text{toi.length} == 3$
- fromi and toi consist of uppercase English letters.
- $\text{fromi} \neq \text{toi}$