

332 . Reconstruir Itinerário

Duro 5,6 mil 1,8K

Empresas

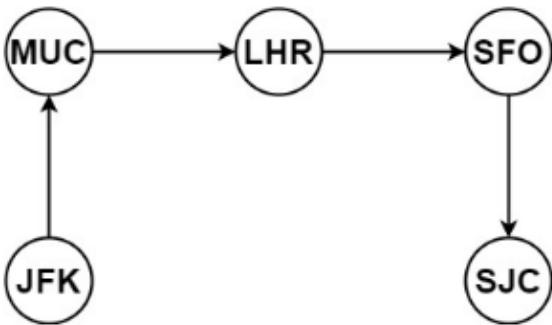
Você recebe uma lista de companhias aéreas `tickets` onde representam os aeroportos de partida e chegada de um voo. Reconstrua o itinerário em ordem e devolva-o. `tickets[i] = [fromi, toi]`

Todos os bilhetes pertencem a um homem que parte de "JFK", portanto, o itinerário deve começar com "JFK". Se houver vários itinerários válidos, você deverá retornar o itinerário que possui a menor ordem lexical quando lido como uma única string.

- Por exemplo, o itinerário ["JFK", "LGA"] tem uma ordem lexical menor que ["JFK", "LGB"].

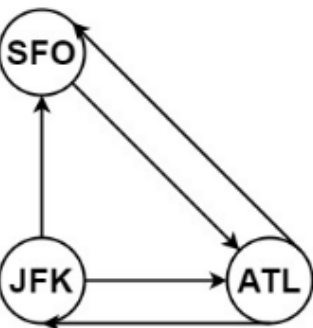
Você pode assumir que todos os bilhetes formam pelo menos um itinerário válido. Você deve usar todos os ingressos uma vez e apenas uma vez.

Exemplo 1:



Entrada: tickets = [["MUC","LHR"],["JFK","MUC"],["SFO","SJC"],["LHR","SFO"]] Saída: ["JFK","MUC","LHR","SFO","SJC"]

Exemplo 2:



Entrada: ingressos = [["JFK","SFO"],["JFK","ATL"],["SFO","ATL"],["ATL","JFK"],["ATL","SFO"]]
Saída: ["JFK","ATL","JFK","SFO","ATL","SFO"]
Explicação: Outra reconstrução possível é ["JFK","SFO","ATL","JFK","ATL","SFO"] mas é maior em ordem lexical.

Restrições:

- `1 <= tickets.length <= 300`
- `tickets[i].length == 2`
- `fromi.length == 3`
- `toi.length == 3`
- `fromi` e `toi` consistem em letras maiúsculas do inglês.
- `fromi != toi`