

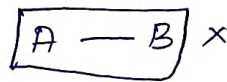
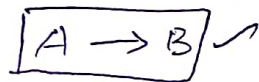
reconstruct itinerary : Hard Sum.

Goal: Start from JFK and cover the route (for all ticket) in lexical order (if we have multiple choices from point).

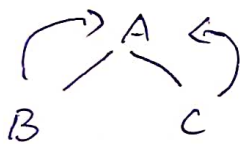
Return the order of travel.

Cases:

1. Ticket are only 1 way. that mean directed graph.



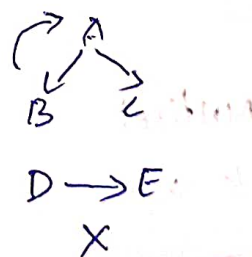
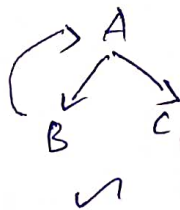
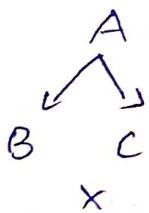
2. Return route with smallest lexical order.



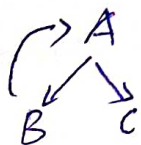
$A \rightarrow \overset{\text{small}}{B} \rightarrow H \rightarrow C \rightarrow A$ ✓

$A \rightarrow C \rightarrow A \rightarrow B \rightarrow A$ ✗

3. All ticket form a valid route. (graph must single component)



4. all ticket use only once.



$A \rightarrow B \rightarrow A \rightarrow C$ ✓

$A \rightarrow B - A \rightarrow B \rightarrow A \rightarrow C$ ✗

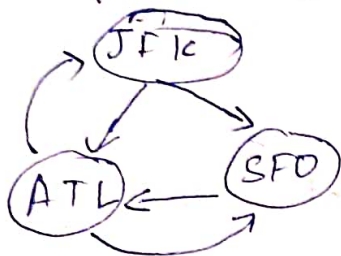
repeat

choice of DS

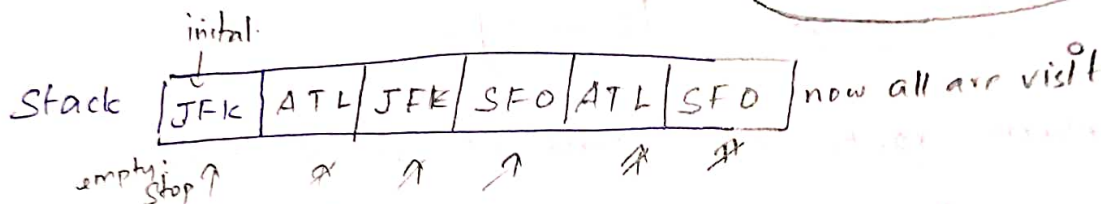
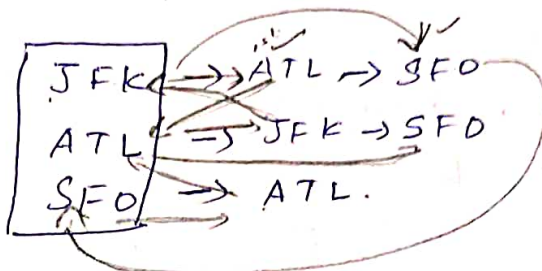
- 1) Multiset \rightarrow keep value arranged
- 2) Map \rightarrow For $O(1)$ search time for given key.
- 3) Stack \rightarrow used to transnet.
- 4) String vector \rightarrow store final answer & reversed it

ex:

Graph



adjust list using unordered map
<string, multiset>

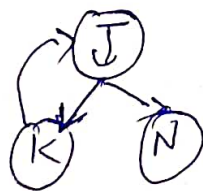
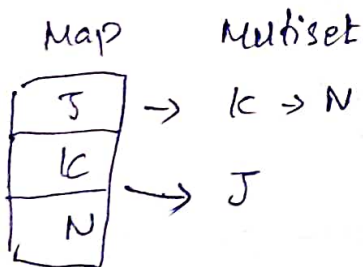


Answer vector : SFO \rightarrow ATL \rightarrow SFO \rightarrow JFK \rightarrow ATL \rightarrow JFK

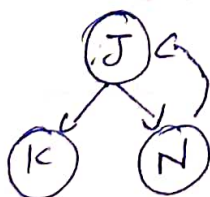
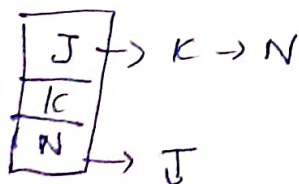
\leftarrow Reverse it

\therefore JFK \rightarrow ATL \rightarrow JFK \rightarrow SFO \rightarrow ATL \rightarrow SFO.

Addition case



$\rightarrow J \rightarrow K \rightarrow J \rightarrow N \checkmark$



But here

\times $J \rightarrow K \rightarrow$ dead state. u cant read right part.

So that now u consider valid root first and then lexical order.

$\therefore J \rightarrow N \rightarrow J \rightarrow K \checkmark$