**Problem :** [**https://leetcode.com/problems/cheapest-flights-within-k-stops/**](https://leetcode.com/problems/cheapest-flights-within-k-stops/)

**Approach :**

-> **In dijkstra , 2 cases needs to be taken care of:**

1. **When a node is popped from the set.**
2. **When entries for a node are added to the set.**

-> Here we need to add a node to set if :

1. If it’s new dist found < old dist , update the minDist vector and optimal steps.
2. if number of steps taken < old no. of steps

-> We need to process a popped node only if :

1. No of steps are lesser than last popped entry for that node ,(bcoz the distances are anyways popped in increasing order, so now we may get an advantage of processing this entry if no of entries are less than the last one).

(Initially the optimal steps always corresponds to entry with optimal distance).

**Code with comments:** [**https://leetcode.com/submissions/detail/556336282/**](https://leetcode.com/submissions/detail/556336282/)