

DAY 19

INTERVIEW BIT PROBLEMS :

1. Reorder List

Given a singly linked list

$L: L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$,

reorder it to:

$L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$

You must do this in-place without altering the nodes' values.

For example,

Given {1,2,3,4}, reorder it to {1,4,2,3}.

CODE :

PYTHON

Definition for singly-linked list.

class ListNode:

def __init__(self, x):

self.val = x

self.next = None

class Solution:

@param A : head node of linked list

@return the head node in the linked list

def reorderList(self, head):

modify the list only if it has more than one node

if not head or not head.next:

return head

find the middle of the list using slow and fast pointers algorithm

previous = None # last node of the left half of the list

slow = fast = head

while fast and fast.next:

prev = slow

slow = slow.next

fast = fast.next.next

prev.next = None # detach left half of the list from the right half

reverse right half of the list

prev = None

curr = slow # slow is the start of the right half

while curr:

link = curr.next

curr.next = prev

prev = curr

curr = link

prev now points to the head of reversed right half

```

# combine left and right half, attach it to dummy node
dummy = ListNode(None)
tail = dummy # current tail of a new list
while head: # left list is always gonna be shorter
    link = head.next # save link to the next node in the left list
    head.next = prev
    tail.next = head # attach connected 2 nodes to the tail
    tail = prev
    head = link # move to the next node in left list
    prev = prev.next
# if the length of the original list was odd, right half is gonna have
# 1 node more than the left half
if prev:
    tail.next = prev
return dummy.next # return new head

```

2. Hotel Bookings Possible

A hotel manager has to process N advance bookings of rooms for the next season. His hotel has K rooms. Bookings contain an arrival date and a departure date. He wants to find out whether there are enough rooms in the hotel to satisfy the demand. Write a program that solves this problem in time $O(N \log N)$.

Input:

First list for arrival time of booking.
 Second list for departure time of booking.
 Third is K which denotes count of rooms.

Output:

A boolean which tells whether its possible to make a booking.
 Return 0/1 for C programs.
 0 -> No there are not enough rooms for N booking.
 1 -> Yes there are enough rooms for N booking.

Example :

Input :

```

Arrivals : [1 3 5]
Departures : [2 6 8]
K : 1

```

Return : False / 0

At day = 5, there are 2 guests in the hotel. But I have only one room.

CODE :

PYTHON

class Solution:

```

    # @param arrive : list of integers

```

```

# @param depart : list of integers
# @param K : integer
# @return a boolean
def hotel(self, arrive, depart, K):
    n=len(arrive)
    out=[]#creating a list which stores both arrival and departure
    for i in range(n):
        out.append((arrive[i],1))
        out.append((depart[i],0))
    out.sort()
    available,maxi=0,0
    for i in range(len(out)):
        if out[i][1]==1:
            available+=1
            maxi=max(available,maxi)
        else:
            available-=1
    return K>=maxi

```