Q Longest Happy string

(3) → happy string

a = , b = , c =

(1) → 'a', 'b', 'c'

(2) X aaa, X bbb, X ccc

(1) → a, b, c

longest possible happy string.

Eg: a = 1, b = 1, c = 7

ccacbcc

To have the longest happy string, we want to greedily do 2 things :-

1. Use as many as possible char with the max count.

2. To achieve (1), we need to have an 'a' in the array as much delimiter

a = 1 b = 1 c = 7 6 5 4 3

c c a c c b c c

Q least number of unique integers after k removals.

array of integers

arr = [4, 3, 1, 1, 3, 3, 2]

k

k = 3

arr = 2

1 3

(1) Count the occurences of each no using Hashmap.

4	1
3	3
1	2
2	1

(2) Using array to count each occurence.

2 1 1

0 1 2 3 4 5

k = 2 1

signifies that there are 2 ele with freq (1)

(3) From small to big, for each unvisited least freq element, deduct from k the multiplication with the no. of ele of same occurence.

→ If reaching 0, then deduct the corresponding unique count

Q Queue reconstruction by height

Greedy approach :

Assume, all people are of same height.

[7, 1], [7, 0]

[7, 0], [7, 1]

[7,1], [6,1], [7,0]

(7,0), [7,1], [6,1]

→ Sort the tallest guys in the ascending order by k- values and then insert them one by one into output queue at the indexes equal to their k- values.

→ Take the next height in decreasing order. Sort the guys of that height in ascending order by k- values and then insert them one by one into output queue.

[7, 0], [4,4], [7, 1], [5, 0], [6, 1], [5, 2]

[7, 0], [7, 1], [6, 1], [5, 0], [5, 2], [4, 4]

[5,0] [7,0], [5, 2], [6, 1], (4, 4), (7, 1)

Try implementation