Question: https://leetcode.com/problems/substring-with-concatenation-of-all-words/

I am personally a bit disappointed with this problem, because there are some constraints which are not mentioned properly, like you have to make an assumption that all string in words array is of same length.

So I will just link the youtube video I have referred to.

Link: https://www.youtube.com/watch?v=Bbu4Qjzf7A0&t=934s

Code:  
class Solution {

public List<Integer> findSubstring(String s, String[] words) {

if(s.length()==0||words.length==0){

return new ArrayList<Integer>();

}

List<Integer> res = new ArrayList<Integer>();

int eachWordLength = words[0].length(), maxLength = words.length\*eachWordLength;

HashMap<String, Integer> wordFrequency = new HashMap<>();

for(String word:words){

wordFrequency.put(word, wordFrequency.getOrDefault(word, 0)+1);

}

for(int i=0; i<=s.length()-maxLength; i++){

HashMap<String, Integer> subFrequency = new HashMap<>();

for(int j=0; j<words.length; j++){

int startIndex = i+j\*eachWordLength;

String sub = s.substring(startIndex, startIndex+eachWordLength);

if(!wordFrequency.containsKey(sub)) break;

subFrequency.put(sub, subFrequency.getOrDefault(sub, 0)+1);

if(subFrequency.get(sub)>wordFrequency.get(sub)) break;

if(j+1==words.length){

res.add(i);

}

}

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>