Question :

Leet code for Search insert position

Solution:

```
class Solution {
    public int searchInsert(int[] nums, int target) {
        int l = 0;
        int r = nums.length;
        while(l<r)
        {
            final int m =(l+r)/2;
            if(nums[m]== target)
            return m;
            if(nums[m]<target)
            l=m+1;
            else
            r=m;
        }
        return l;
    }
}
```

Leet code for Substring with Concatenation of All Words

Solution:

```
class Solution {
    public List<Integer> findSubstring(String s, String[] words) {
        if (s.isEmpty() || words.length == 0)
      return new ArrayList<>();
```

```java
    final int k = words.length;
    final int n = words[0].length();
    List<Integer> ans = new ArrayList<>();
    Map<String, Integer> count = new HashMap<>();

    for (final String word : words)
      count.merge(word, 1, Integer::sum);

    for (int i = 0; i <= s.length() - k * n; ++i) {
      Map<String, Integer> seen = new HashMap<>();
      int j = 0;
      for (; j < k; ++j) {
        final String word = s.substring(i + j * n, i + j * n + n);
        seen.merge(word, 1, Integer::sum);
        if (seen.get(word) > count.getOrDefault(word, 0))
          break;
      }
      if (j == k)
        ans.add(i);
    }

    return ans;


    }
}
```