2020-01-18 Greedy Algorithms

Q1: IsSubSequence

Link: https://leetcode.com/problems/is-subsequence/

Given a string **s** and a string **t**, check if **s** is subsequence of **t**.

You may assume that there is only lower case English letters in both **s** and **t**. **t** is potentially a very long (length ~= 500,000) string, and **s** is a short string (<=100).

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, `"ace"` is a subsequence of `"abcde"` while `"aec"` is not).

Example 1:
s = `"abc"`, **t** = `"ahbgdc"`

Return `true`.

Example 2:
s = `"axc"`, **t** = `"ahbgdc"`

Return `false`.

Q2: Car pooling

Link: https://leetcode.com/problems/car-pooling/

You are driving a vehicle that has `capacity` empty seats initially available for passengers. The vehicle **only** drives east (ie. it **cannot** turn around and drive west.)

Given a list of `trips`, `trip[i] = [num_passengers, start_location, end_location]` contains information about the `i`-th trip: the number of passengers that must be picked up, and the locations to pick them up and drop them off. The locations are given as the number of kilometers due east from your vehicle's initial location.

Return `true` if and only if it is possible to pick up and drop off all passengers for all the given trips.

Example 1:

```
Input: trips = [[2,1,5],[3,3,7]], capacity = 4

Output: false
```

Example 2:

```
Input: trips = [[2,1,5],[3,3,7]], capacity = 5

Output: true
```

Example 3:

```
Input: trips = [[2,1,5],[3,5,7]], capacity = 3

Output: true
```

Example 4:

```
Input: trips = [[3,2,7],[3,7,9],[8,3,9]], capacity = 11

Output: true
```

Q3: Remove K Digits

Link: https://leetcode.com/problems/remove-k-digits/submissions/

Given a non-negative integer *num* represented as a string, remove *k* digits from the number so that the new number is the smallest possible.

Note:

- The length of *num* is less than 10002 and will be ≥ *k*.
- The given *num* does not contain any leading zero.

Example 1:

```
Input: num = "1432219", k = 3

Output: "1219"

Explanation: Remove the three digits 4, 3, and 2 to form the new number 1219 which is
the smallest.
```

Example 2:

```
Input: num = "10200", k = 1

Output: "200"

Explanation: Remove the leading 1 and the number is 200. Note that the output must
not contain leading zeroes.
```

Example 3:

```
Input: num = "10", k = 2

Output: "0"

Explanation: Remove all the digits from the number and it is left with nothing which
is 0.
```

Q4: Rearrange string K distance apart

Given a non-empty string str and an integer k, rearrange the string such that the same characters are at least distance k from each other.

All input strings are given in lowercase letters. If it is not possible to rearrange the string, return an empty string "".

Example:

```
Input: str = "aabbcc", k = 3

Output: "abcabc"

Explanation: The same letters are at least distance 3 from each other.
```