## 1. Arrays and Strings

Main techniques:

* Two pointers
* Sliding window
* Prefix sum

## 2. Hashing

Hash table read/write is always O(1)

* But involves significant overhead, more like O(10)
* Arrays are O(N) for modification of length, but less overhead

Hash tables may take up more space

* Arrays are set to size of data

Hash tables in Python

* Dictionaries: {key1: value1, key2:value2}
  + Unordered
* Sets: {value1, value2}
  + Unordered
  + Does not track frequency, i.e. same number not present more than one time

Size of hash table array (bins) and base of hashing function (k) should be *coprime*

* Coprime: the highest common factor (greatest common divisor) of a set of numbers is 1
* This reduces the frequency of collisions

Example hashing function with base k:

(first char) + k \* (second char) + k^2 \* (third char) + ...