Dictionary Usage

A dictionary in swift is an unordered collection of items that stores its elements as [key : value] pairs. You get it? Like an actual dictionary where the key is the word and the value is its meaning. How clever. They are even more valuable for their speed than they are for their structure. Sometimes you will use a dictionary for its speed alone, storing data as keys and just setting the value to true. In these cases it doesn't matter what the value is, you just need to be able to access its elements in constant time. A good example of this is the leetcode question "Two Sum".

Two Sum Prompt:
Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

This problem can be solved with nested for loops. But that approach is unacceptably slow (On^2).

SOLUTION SPOILER:
A much better solution to the problem is use a dictionary to store the numbers in the array, comparing them with the difference from target number and returning them if they sum to the target

```
func twoSum(_ nums: [Int], _ target: Int) -> [Int] {

        var dict = [Int : Int]()

        for (i, n) in nums.enumerated() {
            var difference = target - n

            if dict[difference] != nil {
                return [dict[difference]!, i]
            }
            dict[n] = i
        }
        return []
    }
```

TLDR:
If you get stuck, try throwing a dictionary at the problem.That’ll probably work.