001 Two Sum

2018年3月25日 13:49

Question:

Given an array of integers, return indices of the two numbers such that they add up to a specific target.
You may assume that each input would have exactly one solution, and you may not use the *same* element twice.
给出一个整数数组，返回其中两个数的索引，使得这两个数之和等于指定的目标值。
可以假定每个输入只有一个解，且不能使用一个元素两次。
Example:
Given nums = [2, 7, 11, 15], target = 9,
Because nums[0] + nums[1] = 2 + 7 = 9,cC
return [0, 1].

来自 <https://leetcode.com/problems/two-sum/description/>

Solution for Python3:

```python
class Solution:
    def twoSum(self, nums, target):
        """
        :type nums: List[int]
        :type target: int
        :rtype: List[int]
        """
        D = {}
        for i, n in enumerate(nums):
            if (target - n) in D:
                return [D.get(target - n), i]
            else:
                D[n] = i
```

Solution for C++:

```cpp
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        map<int, int> map;
        for (int i = 0; i < nums.size(); i++) {
            if (map.find(target - nums[i]) != map.end()) {
                return vector<int> {map[target - nums[i]], i};
            }
            map[nums[i]] = i;
        }
        return vector<int> (2);
    }
};
```

Appendix:

enumerate(iterable, index=0):

1) python内置函数，将一个可迭代对象（如list,str）组成一个索引序列，可同时获得索引和值。返回enumerate对象，该对象的每个元素是不可变元祖tuple(index,value)。
2) 接收第二个参数,索引的起始值，默认为0。

Complexity Analysis:

1) Time complexity : O(n). We traverse the list containing n elements only once. Each look up in the dict costs only O(1) time.
2) Space complexity : O(n). The extra space required depends on the number of items stored in the dict, which stores at most n elements.