Two Sum

* Question Description:

Given an array of integers, find two numbers such that they add up to a specific target number. The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. The answers (index1 and index2) are not zero-based.

* Required Complexity:

O(n)

* Input:

Given nums = [2, 7, 11, 15], target = 9. Assume that each input would have exactly one solution, and you may not use the same element twice.

* Explanation:

nums [0] + nums [1] = 2 + 7 = 9

return [index1, index2] = [0, 1]

* Analyze:

Assume that the values in nums array are all positive.

1. Select low\_bound from input array which equals to 2 and high\_bound from the input array which equals to 15.
2. Compare low\_bound with target:
   1. if target > = low\_bound, continue;
   2. else target < low\_bound, no index pair;
3. Compare the high\_bound with target:
   1. if target >= high\_bound, continue;
   2. else target < high\_bound, high\_bound = target.
4. Create a set from low\_bound to high\_bound. The set is used to store whether the value appears in the input array.

(Non-zero indicates the location of the value in the input array.)

[2, 7]

[0, 1]

which stands for:

set [2] = 0, set [7] = 1

1. Scan the set table from scratch to the end.

* First Round:

set [2] = 0, target – 2 = 9 – 2 = 7

set [7] = 1 (exists)

1. return pair (0, 1) as index pair.

* Special Circumstances:

Input array is empty, then returns null as index pair.

* Space Complexity:

O(n)

* Speed Complexity:

O(n)