[**Maximum Size Subarray Sum Equals k**](https://leetcode.com/problems/maximum-size-subarray-sum-equals-k/)

**import** java.util.HashMap;

**public** **class** MaximumSubArrayEqualsK {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int**[] nums = {1, -1, 5, -2, 3};

System.***out***.println(*maxSubArrayLen*(nums , 3));

}

**public** **static** **int** maxSubArrayLen(**int**[] nums, **int** k) {

**if**(nums == **null** || nums.length == 0) {

**return** 0;

}

HashMap<Integer , Integer> map = **new** HashMap<>();

map.put(0, -1);

**int** max = Integer.***MIN\_VALUE***;

**int** sum = 0;

**for**(**int** i = 0 ; i < nums.length ; i++) {

sum += nums[i];

**if**(map.containsKey(sum - k)) {

max = Math.*max*(max, i- map.get(sum - k));

}

**if**(!map.containsKey(sum)) {

map.put(sum, i);

}

}

**return** max == Integer.***MIN\_VALUE*** ? 0 : max;

}

}

Time Complexity : O(n) , n is length of given num array

Space Complexity : O(n), n is length of given num array