Add Two Numbers (/problems/add-two-numbers/)

Submission Detail

1563 / 1563 test cases passed.

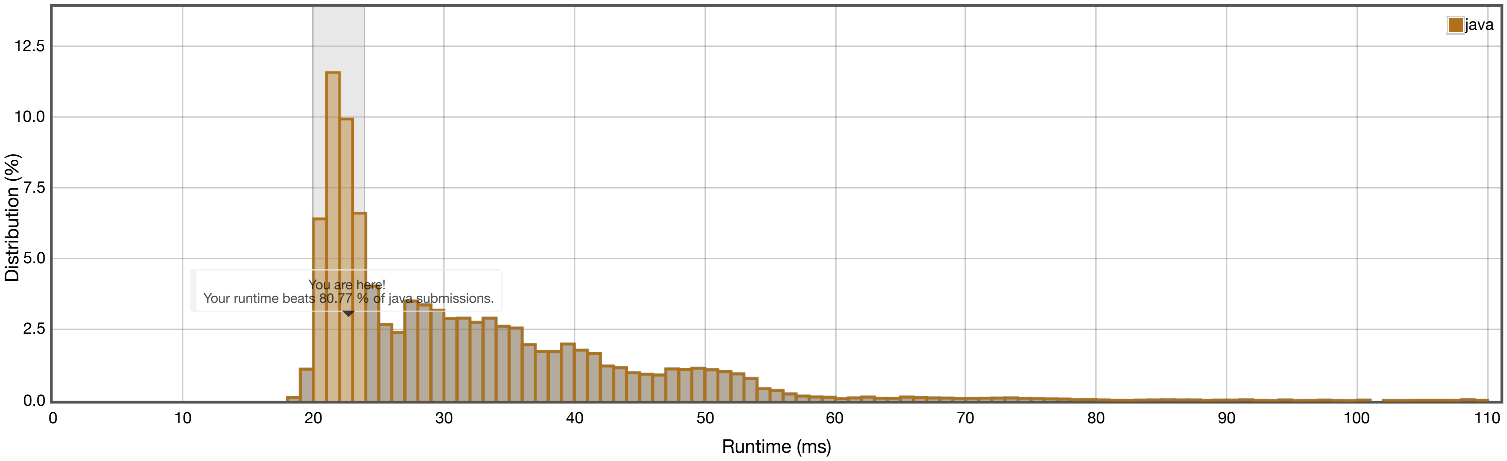
Status: Accepted

Runtime: **22 ms**

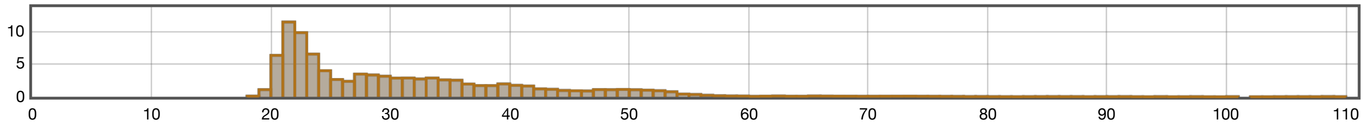
Memory Usage: **29.1 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

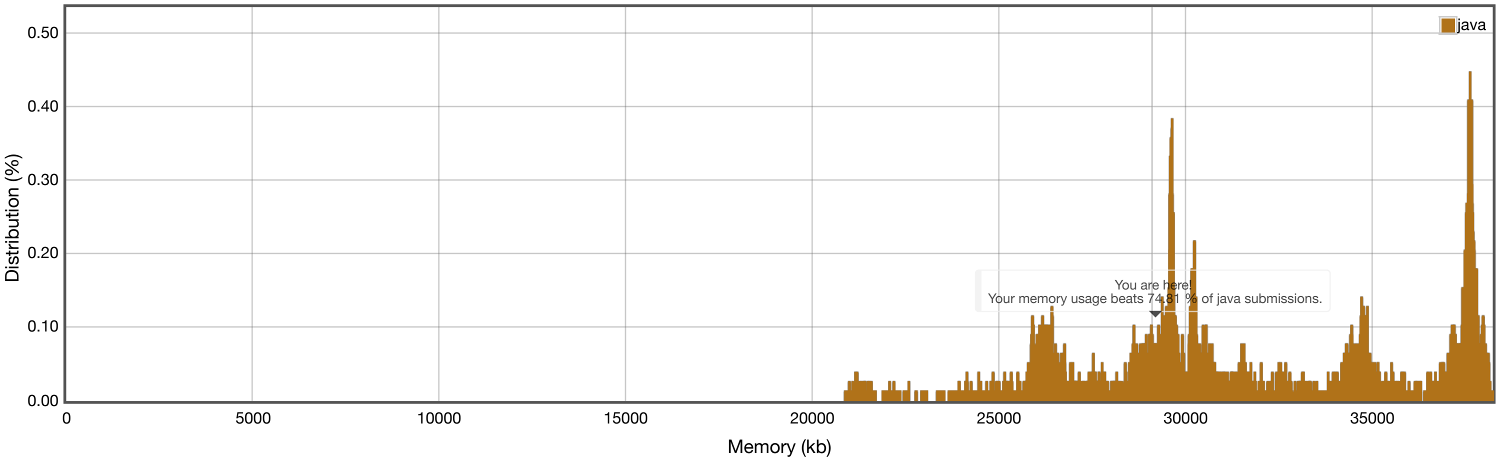


You are here!
Your runtime beats 80.77 % of java submissions.

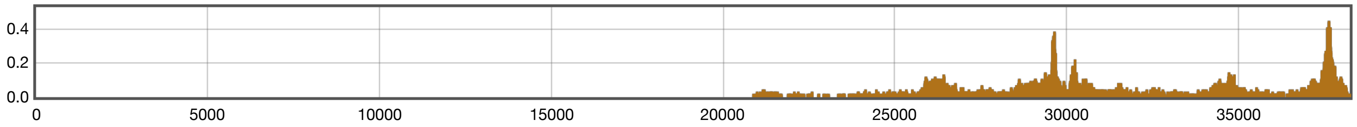


Zoom area by dragging across this chart

Accepted Solutions Memory Distribution



You are here!
Your memory usage beats 74.81 % of java submissions.



Zoom area by dragging across this chart

Invite friends to challenge **Add Two Numbers**

130

Submitted Code: 0 minutes ago

Language: java

Edit Code

```java
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) { val = x; }
 * }
 */
class Solution {
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

    ListNode anshead = null;
     ListNode temp = null;
    int carry = 0;
    int ans = 0;

        while(l2 != null || l1 != null || carry > 0)
        {
            //System.out.println("dd"+carry);


            int tmp = 0;
            if(l2 != null && l1 != null)
                tmp = l1.val + l2.val + carry;
            else
            if( l1 != null)
                tmp = l1.val + carry;
            else
            if(l2 != null)
                tmp = l2.val + carry;
            else
                tmp = carry;

            ans = tmp%10;
            carry = tmp / 10;

            temp = new ListNode(ans);
            temp.next = anshead;
            anshead = temp;

            if( l1 != null)
                 l1 = l1.next;
            if(l2 != null)
                l2 = l2.next;

            }

    return reverseNode(anshead);
    }


public static ListNode reverseNode(ListNode head)
    {
        // Initialize current, previous and
        // next pointers
        ListNode current = head;
        ListNode prev = null, next = null;


        while (current != null)
        {
            // Store next
            next = current.next;

            // Reverse current node's pointer
            current.next = prev;

            // Move pointers one position ahead.
            prev = current;
            current = next;
        }
        head = prev;

    return head;
    }

```

Back to problem (/problems/add-two-numbers/)