|  |  |
| --- | --- |
| **ALGORITHM** | **MINIMIUM OPERATIONS TO REDUCE X TO ZERO** |
| INPUT: | Integer array nums & Integer x.  (**int**[] nums, **int** x) |
| OUTPUT: | Result of minimum number of operations to reduce x to exactly 0 if it's possible otherwise return -1 |
|  |  |
| STEP 1: | START |
| STEP 2: | Subtract the total number of nums from the value of x  And save it in a variable target  **int** target = Arrays.*stream*(nums).sum() - x; |
| STEP 3: | Initialize the nums.length and store it in variable n  Initialize the numOperation = -1  Initialize the current = 0  **int** n = nums.length, numOperation = -1, current = 0; |
| STEP 4: | Iterate through the array of nums starting at rightmost position  **for** (**int** right = 0, left = 0; right < n; right++) |
| STEP 5: | Update the current position as it iterates  current += nums [right] |
| STEP 6: | Continue the iteration:  while (current > x left <= right)  And set the:  current -= nums [left++] |
| STEP 7: | As it iterates Check if below condition is meet and get the numOperation:  **if** (current == target)  numOperation = Math.*max*(numOperation, right - left + 1); |
| STEP 8: | Return numOperation by subtracting nums.length from wSize  else return -1 |
| STEP 9: | STOP |