**7 kyu**

**Minimum Steps (Array Series #6)**

12988% of 15077 of608[MrZizoScream](https://www.codewars.com/users/MrZizoScream)

C++

* [TRAIN](https://www.codewars.com/kata/minimum-steps-array-series-number-6/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/minimum-steps-array-series-number-6/solutions/cpp)

[Forks (1)](https://www.codewars.com/kata/minimum-steps-array-series-number-6/forks/cpp)

[Discourse (104)](https://www.codewars.com/kata/minimum-steps-array-series-number-6/discuss/cpp)

* |
* Share this kata:

**Task**

***Given*** *an array of N integers, you have to find* ***how many times*** *you have to* ***add up the smallest numbers****in the array until* ***their Sum*** *becomes greater or equal to* ***K***.

**Notes:**

* ***List size*** is *at least 3*.
* ***All numbers*** *will be* ***positive***.
* ***Numbers*** could *occur more than once* , ***(Duplications may exist)***.
* Threshold ***K*** will *always be reachable*.

**Input >> Output Examples**

1- minimumSteps({1, 10, 12, 9, 2, 3}, 6) ==> return (2)

***Explanation***:

* We *add two smallest elements* ***(1 + 2)***, *their sum is 3* .
* ***Then*** we ***add the next smallest number to it (3 + 3)*** , so *the sum becomes 6* .
* ***Now*** the result is greater or equal to ***6*** , *Hence the output is (2) i.e (2) operations are required to do this* .

2- minimumSteps({8 , 9, 4, 2}, 23) ==> return (3)

***Explanation***:

* We *add two smallest elements* ***(4 + 2)***, *their sum is 6* .
* ***Then*** we ***add the next smallest number to it (6 + 8)*** , so *the sum becomes 14* .
* ***Now*** we ***add the next smallest number (14 + 9)*** , so *the sum becomes 23* .
* ***Now*** the result is greater or equal to ***23*** , *Hence the output is (3) i.e (3) operations are required to do this* .

3- minimumSteps({19,98,69,28,75,45,17,98,67}, 464) ==> return (8)

***Explanation***:

* We *add two smallest elements* ***(19 + 17)***, *their sum is 36* .
* ***Then*** we ***add the next smallest number to it (36 + 28)*** , so *the sum becomes 64* .
* We need to ***keep doing this*** *until****the sum****becomes greater or equal to****K****(464 in this case)*, which will require ***8*** Steps .

Expected Time ComplexityO(n Log n)

[**Playing with Numbers Series**](https://www.codewars.com/collections/playing-with-numbers)

[**Playing With Lists/Arrays Series**](https://www.codewars.com/collections/playing-with-lists-slash-arrays)

[**For More Enjoyable Katas**](http://www.codewars.com/users/MrZizoScream/authored)

ALL translations are welcomed

Enjoy Learning !!

**Zizou**

<https://www.codewars.com/kata/minimum-steps-array-series-number-6/cpp>

#include <vector>

#include <iostream>

using namespace std;

int minimumSteps (vector <int> numbers, int k)

{

int max = 0;

for(int i =0; i<numbers.size(); i++) {

if(numbers[i] > max) {

max = numbers[i];

}

}

int frec[max + 1];

for(int i =0; i<max+1; i++) {

frec[i] = 0;

}

int times = 0;

for(int i =0; i<numbers.size(); i++) frec[numbers[i]]++;

int sum = 0;

for (int i = 1; i < max + 1 ; i++)

{

while(frec[i] > 0)

{

if(sum < k)

{

sum += i;

frec[i]--;

times++;

}

else

{

break;

}

}

}

if(times-1 <0) return 0;

return times-1;

}

int main() {

/\*

int arr[] = { 1, 10, 12, 9, 2, 3 }; //2

int n = sizeof(arr)/sizeof(int);

vector<int> v;

for(int i =0; i<n; i++) {

v.push\_back(arr[i]);

}

cout << minimumSteps(v, 6) << endl;

\*/

int arr[] = { 1,1,1};

int n = sizeof(arr)/sizeof(int);

vector<int> v;

for(int i =0; i<n; i++) {

v.push\_back(arr[i]);

}

cout << minimumSteps(v, 3) << endl;

return 0;

}