



Jesse and Cookies



by vatsalchanana

Problem

Submissions

Leaderboard

Discussions

Jesse loves cookies. He wants the sweetness of all his cookies to be greater than value K . To do this, Jesse repeatedly mixes two cookies with the least sweetness. He creates a special combined cookie with:

$\text{sweetness} = (1 \times \text{Least sweet cookie} + 2 \times \text{2nd least sweet cookie})$.

He repeats this procedure until all the cookies in his collection have a sweetness $\geq K$.

You are given Jesse's cookies. Print the number of operations required to give the cookies a sweetness $\geq K$. Print -1 if this isn't possible.

Input Format

The first line consists of integers N , the number of cookies and K , the minimum required sweetness, separated by a space.
The next line contains N integers describing the array A where A_i is the sweetness of the i^{th} cookie in Jesse's collection.

Constraints

$$1 \leq N \leq 10^6$$

$$0 \leq K \leq 10^9$$

$$0 \leq A_i \leq 10^6$$

Output Format

Output the number of operations that are needed to increase the cookie's sweetness $\geq K$.

Output -1 if this isn't possible.

Sample Input

```
6 7
1 2 3 9 10 12
```

Sample Output

```
2
```

Explanation

Combine the first two cookies to create a cookie with $\text{sweetness} = 1 \times 1 + 2 \times 2 = 5$

After this operation, the cookies are **3, 5, 9, 10, 12**.

Then, combine the cookies with sweetness **3** and sweetness **5**, to create a cookie with resulting $\text{sweetness} = 1 \times 3 + 2 \times 5 = 13$


Now, the cookies are **9, 10, 12, 13**.

All the cookies have a sweetness ≥ 7 .

Thus, **2** operations are required to increase the sweetness.

Difficulty: Easy

Rate This Challenge:

[More](#)Current Buffer (saved locally, editable)  

C++



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <queue>
5 #include <iostream>
6 #include <algorithm>
7 using namespace std;
8
9 const bool DEBUG = false;
10
11 int main() {
12     long N;
13     cin >> N;
14     long K;
15     cin >> K;
16     priority_queue<long, vector<long>, greater<long> > A;
17     for (size_t i=0; i<N; ++i) {
18         int t;
19         cin >> t;
20         A.push(t);
21     }
22
23     if (A.empty()) {
24         cout << "-1" << endl;
25         return -1;
26     }
27
28     long counter = 0;
29     while (!A.empty()) {
30         long c1 = A.top();
31         if (DEBUG) cout << c1 << endl;
32         if (c1 >= K) {
33             cout << counter << endl;
34             return 0;
35         }
36         A.pop();
37
38         if (A.empty()) {
39             cout << "-1" << endl;
40             return -1;
41         }
42
43         long c2 = A.top();
44         A.pop();
45
46         A.push(c1 + 2*c2);
47         counter++;
48     }
49
50     return 0;
51 }
52
```

Line: 9 Col: 25

 [Upload Code as File](#)☐ Test against custom input[Run Code](#)[Submit Code](#)

Congrats, you solved this challenge!

- | | | |
|-----------------|-----------------|-----------------|
| ✓ Test Case #0 | ✓ Test Case #1 | ✓ Test Case #2 |
| ✓ Test Case #3 | ✓ Test Case #4 | ✓ Test Case #5 |
| ✓ Test Case #6 | ✓ Test Case #7 | ✓ Test Case #8 |
| ✓ Test Case #9 | ✓ Test Case #10 | ✓ Test Case #11 |
| ✓ Test Case #12 | ✓ Test Case #13 | ✓ Test Case #14 |
| ✓ Test Case #15 | ✓ Test Case #16 | ✓ Test Case #17 |
| ✓ Test Case #18 | ✓ Test Case #19 | ✓ Test Case #20 |
| ✓ Test Case #21 | ✓ Test Case #22 | ✓ Test Case #23 |
| ✓ Test Case #24 | ✓ Test Case #25 | ✓ Test Case #26 |
| ✓ Test Case #27 | | |

[Next Challenge](#)

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)