**6 kyu**

**The nth smallest integer**

18288% of 132168 of534[kaminobenzene](https://www.codewars.com/users/kaminobenzene)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/the-nth-smallest-integer/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/the-nth-smallest-integer/solutions/cpp)

[Forks (1)](https://www.codewars.com/kata/the-nth-smallest-integer/forks/cpp)

[Discourse (46)](https://www.codewars.com/kata/the-nth-smallest-integer/discuss/cpp)

* Add to Collection
* |
* Share this kata:

Given a list of integers, return the nth smallest integer in the list. **Only distinct elements should be considered** when calculating the answer. n will always be positive (n > 0)

If the nth small integer doesn't exist, return -1 (C++) / None (Python) / nil (Ruby) / null (JavaScript).

Notes:

* "indexing" starts from 1
* huge lists (of 1 million elements) will be tested

Examples

small({1, 3, 4, 5}, 7) -> -1 // n is more than the size of the list

small({4, 3, 4, 5}, 4) -> -1 // 4th smallest integer doesn't exist

small({45, -10, 4, 5, 4}, 4) -> 45 // 4th smallest integer is 45

If you get a timeout, just try to resubmit your solution. However, if you ***always*** get a timeout, review your code.

<https://www.codewars.com/kata/the-nth-smallest-integer/cpp>

1. #include <iostream>
2. #include <stdio.h>
3. #include <algorithm>
4. #include <vector>
5. #include <set>
7. using namespace std;
9. static long small(std::vector<long> list, long n){
11. //if(n >= list.size())  return -1;
13. std::set<long> s;
15. for(int i =0; i < list.size(); i++)  s.insert(list[i]);
17. std::vector<long> unicos;
18. for(std::set<long>::iterator it = s.begin(); it != s.end(); it++)  unicos.push\_back(\*it);
19. if(n > unicos.size()) return -1;
21. std::sort(unicos.begin(), unicos.end());
22. return unicos[(int)n - 1];
24. }
26. int main() {
28. long arr[] ={455555,2222222,3333333,9879799,79977979,79977979,79977979,79977979,79977979,79977979,79977979,79977979};
30. int size = sizeof(arr)/sizeof(int);
31. std::vector<long> v;
33. for(int i =0; i < size; i++)
34. {
35. v.push\_back(arr[i]);
36. }
38. cout << small(v, 6) << endl;

41. return 0;
42. }