**SOLUTION**

class FirstUnique {

private:

queue<int> q;

unordered\_map<int, int> count;

public:

FirstUnique(){

ios::sync\_with\_stdio(false);

std::cin.tie(nullptr);

std::cout.tie(nullptr);

}

FirstUnique(vector<int>& nums) {

for(int num: nums){

count[num]++;

q.push(num);

}

}

int showFirstUnique() {

while(!q.empty() && count[q.front()] > 1){

q.pop();

}

if(q.empty()){

return -1;

}

else{

return q.front();

}

}

void add(int value) {

if(++count[value] == 1){

q.push(value);

}

}

};

**TIME COMPLEXITY: O(1)**

**SPACE COMPLEXITY: O(N)**