Lin Guohao **1. assignment/6. task** March 2020
Linguohao111@gmail.com
Group 1

Task

Implement the set type which contains integers. Represent the set as a sequence of its elements. Implement as methods: inserting an element, removing an element, returning whether the set is empty, returning whether the set contains an element, returning a random element without removing it from the set, returning the number of even numbers in the set (suggestion: store the number of even numbers and update it when the set changes), printing the set.

Diagonal set type

Set of values

Set(n) = { $a \in Z^n$ I $\forall i \in [1..n]$: $a[i] \in Z \wedge \forall i,j \in [1..n]$: ∄i≠j → a[i] = a[j] }

Operations

1. Inserting an element
Insert an element into the set. (a :Set ,e:Z) → a:=add(a,e)

Formally: *A : a(n):Set , e: Z*
Pre: (a(n) = a' a(n) ∧ e = e')
Post:(Pre∧(∀*i ∈[1..n]:a[i] ≠e)* → *a:=a* ⊕ *e*)

This operation needs any action only if ∀*i ∈[1..n]:a[i] ≠e* otherwise Output "Element already exists" message.

2.Removing an element

Delete an element that exists in the set. (a :Set ,e:Z) → a:=rem(a,e)
Formally: A : a(n):Set , e: Z , cond = (n=0)
Pre: (a(n) = a' a(n) Ù e = e')cond=False
Post:(PreÙ cond=FalseÙ ("i∈[1..n]: ∃a[i]= e) → a:=a ⊖ e)

This operation needs any action only if ∀*i ∈[1..n]:a[i] =e* otherwise Output " Element does not exist" message.

3.*Determine if Set is empty*

Determine if the number of elements in the Set is zero.（a:Set,l:L) → l:=isEmpty(a)
Formally: *A : a(n):Set, l=False*
Pre: (a(n) = a'a(n))

Post:(Pre∧ n = 0 : *l = True*)

This operation will output : l = False "Set is not empty" , l = True "Set is empty".

4.Whether there are specified elements in the Set

The user gives an element, and the program determines whether the element exists in the Set.（a:Set,l:L,e:Z) → l:=isExist(a,e)

Formally: *A : a(n):Set, e:Z,l=False*
Pre: (a(n) = a'a(n),e = e')

Post:(Pre∧∀*i ∈[1..n]:* ∃*a[i] = e* , *l=True*)

This operation will output : When l = False output "Element does not exist" , when l ＝True output "Successfully removed element".

5.Returning a random element

Returns a random element in the Set. (a:Set,e:Z) → e: = ranEle(a);

Formally: *A : (a(n):Set, e:Z)*
Pre: (a(n) = a'a(n))

Post:(Pre∧ *i ∈[1..n](n⩾1): a[i] = e*)

This operation needs any action only if n ⩾1 otherwise Output " Set is empty " message.

6.Count even numbers in set

Count how many even numbers are in a Set.(a:Set,Cnt:Z) → Cnt:= Evnum(a)

Formally: *A : (a(n):Set, cnt:Z) cond :[1..n]* → L cond = ∀*i ∈[1..n],a[i] mod 2 = 0;*
Pre: (a(n) = a'a(n),cnt=0)

Post:(Pre∧ ($cnt = \sum_{cond(i)}^{n} 1$))

This operation needs any action only if n ⩾1 otherwise Output " Set is empty " message.

7.Print Set out.

Print all elements in the Set. (a:Set,e:Z) → Prout(a);

Formally: A : (a(n):Set,e:Z)
Pre: (a(n) = a'a(n))
Post:(Pre∧ ∀ *i ∈[1..n](n⩾1):e= a[i]*)

This operation needs any action only if n ⩾1 otherwise Output " Set is empty " message.

Representation

Only unique elements will be stored

a=

a_1	a_2	a_3	a_n

« $v = < a_1 \quad a_2 \quad a_3 \quad a_n >$

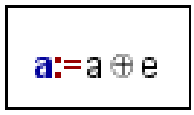
Only a one-dimension array (*v*) is needed, with the help of which any entry of the Set can be get.

.

Implementation[I]

1.Inserting an element

Inserting an element into to the Set , (a :Set ,e:Z) → a:=add(a,e) where the Set is represented by a, Set has at least 0 elements.



2.Removing an element

Remove an existing element in the Set , (a :Set ,e:Z) → a:=rem(a,e) ,The element must exist. The Set is represented by a, n ≥ 1, and n stands for the size of the Set can be implemented as

_____*n=0*_____*L*	
SKIP	a:=a ⊖ e

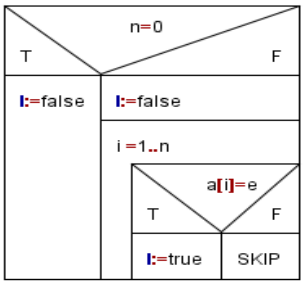
3.*Determine if Set is empty*

Determine if the Set is empty，(a:Set,l:L) → l:=isEmpty(a)，The Set is represented by a, and n stands for the size of the Set can be implemented as

_____*n=0*_____*L*	
True	False

4.Whether there are specified elements in the Set

Determine if the given element exists in the Set,（a:Set,l:L,e:Z) → l:=isExist(a,e), Set must already have elements(n≥1).The Set is represented by a, e is the element given by the user for judgment, and n stands for the size of the Set can be implemented as



5.Returning a random element

Pick one randomly from all the elements of the Set，(a:Set,e:Z) → e: = ranEle(a); Set must already have elements(n≥1).The Set is represented by a, e is the random element returned after

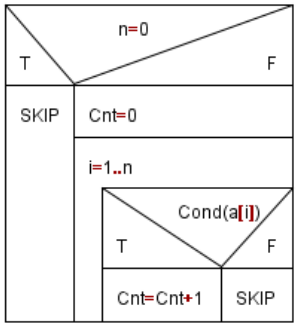
[I] To implement an operation, a program has to be given (not necessarily structogram).

running the method, and n stands for the size of the Set can be implemented as

\ _____ $n=0$ _____ L	
$SKIP$	$e:=RANDOM_1^n a(1..n)$

6.Count even numbers in set

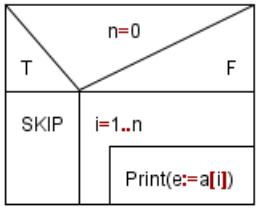
Count the number of even elements in a Set， .(a:Set,Cnt:Z) → Cnt:= Evnum(a); Set must already have elements(n⩾1).The Set is represented by a, Cnt is used to count, and n stands for the size of the Set can be implemented as



cond = $\forall i \in [1..n], a[i] \bmod 2 = 0;$

7.Print Set out.

Print all elements in the Set， (a:Set) → Prout(a); Set must already have elements(n⩾1).The Set is represented by a, e is the element being printed, and n stands for the size of the Set can be implemented as



.

Class

The Set type is worked out as a class. We can access integer types by initializing a vector. At the operations, We need to check if there are the same elements in the Set, otherwise the program should throw an exception.

Diag	Exceptions
-v : int[0 .. n-1] -n : int + insert(int,v): v { query } + remove(v, int) : v { query } + isExist(int, v) : bool { query } + isEmpty(v) : bool + Randnum (v) : int + evennum(v) : int + print(v):v	+ ELEMENT ALREADY EXISTS + SET IS EMPTY + ELEMENT DOES NOT EXIST

The entries in the Set can be represented as a vector<int> or as a dynamic array. In the first case, a vector should be initialized when the object is instantiated.

This class does not need to refactor any operators and functions, and only need to operate on Interger types and vector types.

The class is extended by read and write methods.

For error handling, three exceptions are defined. When the Set is empty and trying to print, determine whether the element exists, try to remove the element, try to return the number of even in the Set, and try to return a random number in the Set, throw a "Set is empty" exception. "Element already exists" exception is thrown when trying to insert the same element. When trying to remove a non-existing element, throw an "Element does not exist" exception.

.

Testing

Testing the operations (black box testing)

1) Insert element
   a) Insert 5
   b) Insert 5 again to determine if an error is reported
2) Determine if Set is empty
   a) Judge directly after instantiating an object
   b) Judge again after inserting random elements
3) Element removal
   a) Try to remove random elements directly after instantiating the object. (Whether to report an error)
   b) Trying to remove another number after inserting a specific number. (Whether to report an error)
   c) Trying to remove a specific number after inserting it.
4) Determine if a specific element exists
   a) Trying to directly determine whether the element exists after instantiating the object.
   b) Trying to determine whether another element exists after inserting a specific element.
   c) Try to determine whether the element exists after inserting it.
5) Give random number test
   a) Try to give a random number directly after instantiating the object (Whether to report an error)
   b) Insert a specific element to determine whether the given random number is this element
6) "Give the count of even numbers in the Set" test
   a) Try to find the count directly after instantiating the object (Whether to report an error)
   b) Check if the count is 0 after inserting an odd number
   c) Check if the count is 1 after inserting an even number
7) Print test
   a) Check print the empty set whether it gives an error