Munkhtenger Munkh-Aldar
WSR292
wsr292@inf.elte

1. assignment/6. task 20th March 2022

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.

Set type

Set of integers

$Set(n) = \{ a \in \mathbb{Z}^n , n \in \mathbb{N} \mid \forall i \in [1..n], \forall j \in [1..n] : i \neq j \rightarrow a[i] \neq a[j] \}$

Operations

1. Inserting an element
Inserting an element to the end of the set if the element is not making any duplicates in the set.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{e}{\mathbb{Z}}$

$Pre = (a=a' \wedge e=e')$

$Post = (\forall i \in [1..n] : a[i] \neq e \rightarrow a := a \oplus \{e\})$

This operation throws an exception if there exists an element from the set which is equal to the given inserting element ($\exists i \in [1..n] : a[i] = e$), otherwise the element should be inserted at the end of the set.

2. Removing an element
Removing an element if the given element exists in the set. If it exists, the last (right-most) element at the end of the set shifts to the removed element's index.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{e}{\mathbb{Z}} \times \underset{l}{\mathscr{L}} \times \underset{ind}{\mathbb{N}^+}$

$Pre = (a=a' \wedge e=e')$

$Post = ((l, ind) = SEARCH_{i=1..n}\, a[i] = e$
$(l, ind) \rightarrow a := a[1..ind\text{-}1] \oplus a[n] \oplus a[ind+1..n\text{-}1])$

This operation throws an exception if the want to be removed element does not exist in the set $\forall i \in [1..n] : a[i] \neq e$, otherwise the last element would shift to the want to be removed element's index.

3. Check if the set is empty
Check if the set has no elements.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{l}{\mathscr{L}}$

$Pre = (a=a')$

$Post = (Pre \wedge l := (a=\{\}))$

If the set is empty, the logical variable returns ‘true’, otherwise it would return ‘false’.

4. Check if the set contains an element
Checking whether the set contains an element. If the element exists in the set, it stores the index of the element.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{e}{\mathbb{Z}} \times \underset{l}{\mathbb{L}} \times \underset{ind}{\mathbb{N}^+}$

$Pre = (a=a' \wedge e=e')$

$Post = (\text{Pre} \wedge (l, ind) = SEARCH_{i=1..n}\, a[i] = \text{e})$

If the element is not in the set, the logical *exists* variable sets to ‘false’, otherwise it sets to ‘true’ and stores the index of the element.

5. Get a random element without removing from the set
Get a random element without removing it from the set.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{e}{\mathbb{Z}}$

$Pre = (a=a')$

$Post = (\text{Pre} \wedge \exists i \in [1..n] : \text{e} := a[i])$

This operation throws an exception if the set is empty, otherwise a random element from the set would return.

6. The number of even numbers in the set.
A count of even numbers in the set.

Formally: $A = \underset{a}{\mathbb{Z}^*} \times \underset{cnt}{\mathbb{N}}$

$Pre = (a=a')$

$Post = (\text{Pre} \wedge \text{cnt} := \sum_{\substack{i=1 \\ 2 \mid a[i]}}^{n} 1)$

This operation returns number 0 if the set is empty or there is no even element in the set, otherwise the counter variable is increasing by 1 for every even element in the set.

Representation

Only the integers not making any duplication with the other elements in the set has to be stored.

$a = a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_n \qquad \leftrightarrow \qquad v = < a_1 \quad a_2 \quad a_3 \quad a_n >$

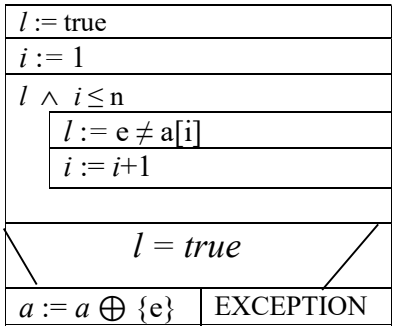
Only a one-dimensional array (v) is needed. In the set, the element which doesn’t make any duplication in the set can be stored.

$$a[i] = \begin{cases} v[i] & \text{if} \quad \forall j \in [1..n] : i \neq j \rightarrow a[i] \neq a[j] \\ \text{EXCEPTION} & \text{if} \quad \exists j \in [1..n] : i \neq j \rightarrow a[i] = a[j] \end{cases}$$

Implementation[1]

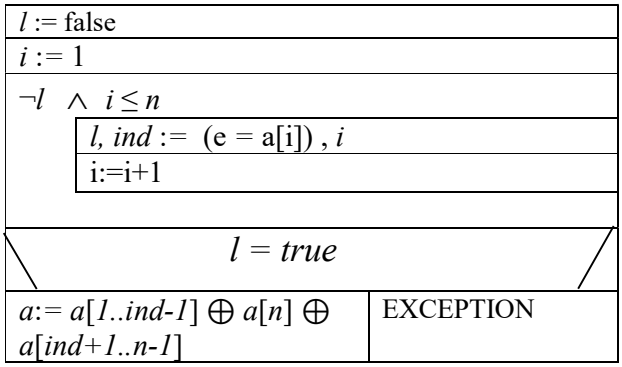
1. Inserting an element

By using optimistic linear search, it has been checked if the element *e* is not equal to all elements of the set. If the element *e* is distinct from each element of the set, the element concatenates to the end of the set. Otherwise, if there is an equal element in the set which is making a duplication, the exception would be thrown.



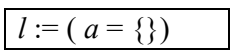
2. Removing an element

By using linear search, it has been checked whether the element to be removed exists or not. If it exists, the set is reconstructed as the last (right-most) element of the set would take place of the to be removed element which its index has been stored in a variable. Hence, exclude the last (right-most) element from the set. Otherwise, if the element does not exist in the set, the exception would be thrown.



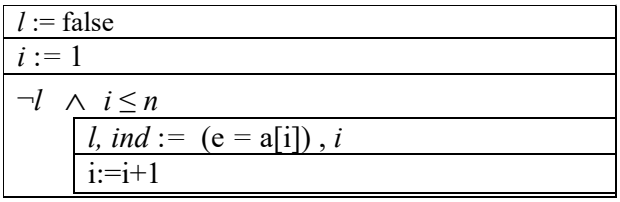
3. Check if the set is empty

The logical variable stores whether the set is equal to an empty set or not. If the set is empty, logical operator returns ‘true’, otherwise it returns ‘false’.



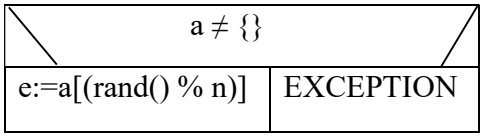
4. Check if the set contains an element

By using linear search, it has been checked whether the set contains a given element. If the element is found in the set, the index of the element is stored in a variable, and it would return ‘true’. Otherwise, if the element has not found in the set, it would return ‘false’.



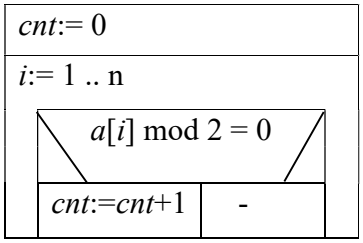
5. Get a random element without removing from the set

If the set is not empty, the index of the random element is generated by using the built-in function rand() ∈ [*1..n*]. Therefore, variable *e* stores the random element from the set by using the generated random index. Otherwise, if the set is empty, exception would be thrown.



6. The number of even numbers in the set.

The counter variable is set to 0 and it is getting incremented by 1 for each even elements in the set. It has been checked if the number is even by whether the element's remainder equals 0 after dividing by 2. Otherwise, if the remainder equals 1 after dividing by 2 for an element, no changes will occur to the counter variable.



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

Testing

Testing the operations (black box testing)

1) Inserting an element
   a) Checking if the inserted element concatenated to the end of the set.
      {1,3,2,6} ⊕ {0} → {1,3,2,6,0}
      {1,3,2,6,0} ⊕ {8} → {1,3,2,6,0,8}
      {1,3,2,6,0,8}⊕ {4} → {1,3,2,6,0,8,4}
   b) Insert an element to the empty set.
      {} ⊕ {1} → {1}

2) Removing an element
   a) Remove an existing element from the set that has 1 element.
      Remove 1 from {1} → {}
   b) Checking if the removed element is replaced by the last (right-most) element of the set and the size of the set has decreased by 1 with excluding the last element. The set has multiple elements and to be removed element is at the first.
      Remove 1 from {1,3,2} → {2,3}
   c) Checking if the removed element is replaced by the last (right-most) element of the set and the size of the set has decreased by 1 with excluding the last element. The set has multiple elements and to be removed element is at the middle.
      Remove 3 from {1,3,2} → {1,2}
   d) Checking if the removed element is replaced by the last (right-most) element of the set and the size of the set has decreased by 1 with excluding the last element. The set has multiple elements and to be removed element is at the last.
      Remove 2 from {1,3,2} → {1,3}

3) Check if the set is empty
   a) Check if the empty set returns ‘true’.
      {} → true
   b) Check if the non-empty set returns ‘false’.
      {1,3,2,6} → false

4) Get a random element without removing from the set
   a) Getting a random element from the set that has 1 element.
      rand() from {1} → 1
   b) Getting a random element from the set that has a multiple element. The rand() function will return a random index from the set. The variable would get a random element based on the returned random index.
      rand() from {1,3,2} → 0 || rand() from {1,3,2} → 1 || rand() from {1,3,2} → 2

Testing based on the code (white box testing)

1. Creating an extreme-size set (0, 1000, 2000, 3000…)
2. Generating and catching exceptions.
   a) Inserting an element
      Trying to insert an element that already exists in the set.
      {1,3,2,6} ⊕ {1} → EXCEPTION

b) Removing an element
   - Trying to remove an element from the empty set.
     Remove 9 from {} → EXCEPTION
   - Trying to remove a non-existing element in the set.
     Remove 9 from {1,3,2,6} → EXCEPTION

c) Get a random element without removing from the set
   - Trying to get a random element from an empty set.
     rand() from {} → EXCEPTION

Testing based on the predicted algorithm from the executive specification (grey box testing)

1) Check if the set contains an element
   a) Search for an element from the empty set.
      Search 3 from {} → false
   b) Search for an element that does not exist from the set that has 1 element.
      Search 3 from {1} → false
   c) Search for an element that exists from the set that has 1 element.
      Search 1 from {1} → true, ind=0
   d) Search for an element that does not exist from the set that has a multiple element.
      Search 5 from {1,3,2} → false
   e) Search for an element that exists from the set that has a multiple element. The first element is searched.
      Search 1 from {1,3,2} → true, ind=0
   f) Search for an element that exists from the set that has a multiple element. The middle element is searched.
      Search 3 from {1,3,2} → true, ind=1
   g) Search for an element that exists from the set that has a multiple element. The last element is searched.
      Search 2 from {1,3,2} → true, ind=2

2) The number of even numbers in the set.
   a) Get the number of even numbers in the empty set.
      {} → 0
   b) Get the number of even numbers in the set that has l element, and the element is even.
      {2} → 1
   c) Get the number of even numbers in the set that has l element, and the element is odd.
      {1} → 0
   d) Get the number of even numbers in the set that has multiple elements, and all elements are even.
      {-2,4,6} → 3
   e) Get the number of even numbers in the set that has multiple elements, and all elements are odd.
      {-1,3,5} → 0
   f) Get the number of even numbers in the set that has multiple elements, and all elements are mixed whether even and odd.
      {-1,-4,3,5,8,10} → 3