**Documentation for the first assignment:**

G31R6T

li zhipeng

**Task:**

Implement the block matrix type which contains integers. These are square matrices that can contain nonzero entries only in two blocks on their main diagonal. Let the size of the first and second blocks be b1 and b2, where 1≤b1,b2≤n-1 and b1+b2=n (in the example, b1=4 and b2=5). Don't store the zero entries. Store only the entries that can be nonzero in a sequence or two smaller matrices. Implement as methods: getting the entry located at index (i, j), adding and multiplying two matrices, and printing the matrix (in a square shape). 

**Diagonal matrix type**

**Set of values:**

Diag(n)={aℤn×ni,j[1..n]: i≠j→a[i,j]=0}

**Operations:**

1. Getting an entry:

Getting theentry ofthe ith column and jthrow (i,j[1..n]): e:=a[i,j].

Formally: A:Diag(n) × ℤ× ℤ× ℤ

a i j e

Pre=( a=a’i=i’j=j’i,j[1..n] )Post =( Pree=a[i,j] )Thisoperation needs any action only ifi=j,otherwise the output is zero.

**Operations:**

1. Insert an element into the set:
2. Remove an element from the set:
3. Check if the set is empty:
4. Check whether something is an element of the set or not:
5. Return a random element from the set:
6. Get the sum of all the element in the set:
7. Print all the element in the set:

**Representation:**

Only one-dimension vector is needed.

**Implementation:**

1.Insert an element into the set:

We simply use the push\_back function sequence.push\_back(element);

1. Removing an element from the set:

Here n is the size of the set which can be get by .size() function since we are using vector to implement elements. The element is denoted by “a” here, l is the logical parameter that indicate whether we find the element of interest or not.

|  |
| --- |
| i:=1 , l=false |
| i<=n ∧ ┐l |
| sequence[i]==a |
| l=true,  sequence.erase(sequence.begin()+i) i:=i+1 |

1. Check if the set is empty:

Here we can use the size function for vector since we are using vector to implement the operations

|  |  |
| --- | --- |
| Sequence.size()=0 | |
| true | false |

1. Check whether something is an element of the set or not:

Here we iterate through the sequence and see if there is any element that is same as the element that is given in the parameter list. L is a logical parameter that indicate whether we have found the given element or not. The given element is indicated by “a” here.

|  |
| --- |
| i:=1 , l=false |
| i<=n ∧ ┐l |
| sequence[i]==a |
| l=true i:=i+1 |
| return l |

1. Return a random element from the set:

In this case we just return the first element in the sequence vector.

1. Get the sum of all the element in the set:

we first initialize the variable sum as zero and we start adding up the element as we go through the sequence.

|  |
| --- |
| sum:=0 |
| i:=1..n |
| sum:=sum+sequence[i] |

1. Print all the element in the set:

we go through the sequence and print out each of the element.

|  |
| --- |
| i:=1..n |
| cout:=cout⊕sequence[i] |

**Class:**

the set type is worked out as a class. The operations of the set type are done with the help of the private variable sequence which is a vector type.

|  |
| --- |
| *Diagram* |
| **-**sequence: vector<int> |
| +insert\_(int): void  +remove\_(int): void  +isempty: bool  +ismember(int): bool  +rand: int  +sum: int  +print: void |

The element in the set can be represented as an integer vector.

**Testing:**

*Test\_case1: for insert\_*

1.First, build a new emtpy set.

2.Insert an element and use the rand function to see if the rand function returns the value of the element that we just inserted, since we just built a new set, there should be only one element in the set and rand will naturally return the element that we just inserted.

*Test\_case2: for remove\_*

1. First build a new empty set.
2. Insert and element and use the isempty function to make sure that the set is not empty anymore.
3. Use remove\_function to remove that element and then use the isempty function to check, if the set is empty then the remove\_ function works.

*Test\_case3: for isempty*

1. First build a new empty set.
2. Insert an element into the set and check whether the set is empty or not.
3. Remove the same element from the set and use the isempty function to check whether the set is empty ot not.

(if in the second step the set is not empty and then in the 3rd step the set is empty then it means that the isemtpy function works.)

*Test\_case4: for ismember*

1. First build a new empty set.
2. Insert an element into the set and check if that element is a member of the set using ismember function
3. Remove the same element from the set and check again if that element is a member of the set.

(if in the second step the ismember function return true and in the 3rd step returns false then it means that ismember function works fine.)

*Test\_case4:for rand*

1. First build a new empty set.
2. We insert 4 element into the set.
3. We use rand function to see whether it returns the first element that we inserted, since in the program the rand function always returns the first element of the vector sequence so here it should also return the first element of the sequence.

*Test\_case6:for sum*

1. First we build a new empty set.
2. We insert 4 elements into the set and then we calculate the sum manually.
3. We use the sum function and check if the result is the same as the sum of the 4 number that we just inserted. If it is equal, then the sum function works.