**YOUNG TABLEAU ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 3 | 4 | 5 |
| 8 | 9 | 12 | 14 |
| 16 | ∞ | ∞ | ∞ |
| ∞ | ∞ | ∞ | ∞ |



2)Y[1,1]= smallest element and Y[1,1]= ∞.

* If Y[1,1]= ∞ then all the positions in Y should be infinity, following the young tableaus rule that the element to the right of a particular number and element below a particular number are always greater than the element in the given position. Therefore Y is empty.

Y[m,n] < ∞.

* Y[m,n] is the last element in the young tableau and following the property of young tableaus all elements to its left and all elements above it are less than this element. Hence Y is full.

PROGRAM:

**public** **static** **void** main(String[] args)

{

Scanner reader = **new** Scanner(System.***in***);

System.***out***.println("Enter the value of m: ");

**int** mRow = reader.nextInt();

System.***out***.println("Enter the value of n: ");

**int** nCol = reader.nextInt();

System.***out***.println(" enter the array size: ");

**int** arraySize = reader.nextInt();

System.***out***.println("Enter the elements of array:");

**int** [] array\_initial = **new** **int**[arraySize];

**int** [][] arr\_ele = **new** **int**[mRow][nCol];

**for**(**int** i=0;i<arraySize;i++)

array\_initial [i]=reader.nextInt();

**int** k=0;

**for**(**int** i=0;i<mRow;i++)

**for**(**int** j=0;j<nCol;j++)

{

**if** (k>= arraySize)

arr\_ele[i][j]=Integer.***MAX\_VALUE*** ;

**else**

{

arr\_ele[i][j]= array\_initial[k];

k++;

}

}

*extract\_minimum*(arr\_ele,mRow,nCol);

**int** [][] array\_heap = **new** **int**[mRow][nCol];

array\_heap=*arrayHeaping*(arr\_ele,0,0,mRow,nCol);

**int** [][] array\_insert = **new** **int**[mRow][nCol];

System.***out***.println("Enter the number to be inserted: ");

**int** x = reader.nextInt();

array\_insert=*array\_inserting*(array\_heap,x,mRow,nCol);

System.***out***.println("Enter the search number: ");

**int** search\_key = reader.nextInt();

*finding\_element*(array\_insert, search\_key,mRow,nCol);

reader.close();

}

**public** **static** **void** extract\_minimum(**int** arr[][],**int** mCount,**int** nCount)

{

System.***out***.println("minimum element is " + arr[0][0]);

arr[0][0]=Integer.***MAX\_VALUE***;

**int** [][] arrheapify = **new** **int**[mCount][nCount];

arrheapify= *arrayHeaping*(arr,0,0,mCount,nCount);

}

**public** **static** **int**[][] arrayHeaping(**int** arr[][],**int** i,**int** j,**int** mCount, **int** nCount)

{

**int** x;

**if**((i==mCount-1)||(j==nCount-1))

**return** arr;

**if**((i<mCount)&&(j<nCount))

{

**if** ((arr[i][j+1]<arr[i][j])||(arr[i+1][j]<arr[i][j]))

{

**if**(arr[i][j+1]<arr[i+1][j])

{

x=arr[i][j+1];

arr[i][j+1]=arr[i][j];

arr[i][j]=x;

*arrayHeaping*(arr,i,j+1,mCount,nCount);

}

**else**

{

x=arr[i+1][j];

arr[i+1][j]=arr[i][j];

arr[i][j]=x;

*arrayHeaping*(arr,i+1,j,mCount,nCount);

}

}

}

**return** arr;

}

**public** **static** **int**[][] array\_inserting(**int** arr[][],**int** k,**int** mCount, **int** nCount)

{

**int** i,j,x;

**if**(arr[mCount-1][nCount-1]!=Integer.***MAX\_VALUE***)

{

System.***out***.println("Young Tableau is full");

**return** arr;

}

arr[mCount-1][nCount-1]= k;

i=mCount-1;

j=nCount-1;

**while**((i>0)&&(j>0))

{

**if**((k>arr[i-1][j]) && (k>arr[i][j-1]))

**return** arr;

**else**

**if**((k<arr[i-1][j])||(k<arr[i][j-1]))

{

**if**(arr[i][j-1]<=arr[i-1][j])

{

x=arr[i-1][j];

arr[i-1][j]=arr[i][j];

arr[i][j]=x;

i--;

}

**else**

{

x=arr[i][j-1];

arr[i][j-1]=arr[i][j];

arr[i][j]=x;

j--;

}

}

}

System.***out***.println("The element is inserted at arr["+ i + "][" + j+"]");

**return** arr;

}

**public** **static** **void** finding\_element(**int** arr[][],**int** k,**int** mCount,**int** nCount)

{

**int** i,j;

i=mCount-1;

j=0;

**while**((i>=0)&&(j<nCount))

{

**if**(arr[i][j]>k)

{

i--;

**if**((i==-1)&&(j==-1))

System.***out***.println("The search element is not found");

}

**else** **if**(arr[i][j]<k)

{

j++;

**if**((i==mCount)&&(j==nCount))

System.***out***.println("The search element is not found");

}

**else**

{

System.***out***.println("The element is found at arr[" + i + "][" + j + "]");

**break**;

}

}

}

}

In the above program these are the functions:

1. “extract\_minimum” function

The recurrence relation is

T(p)=T(p−1) + O (1)

=T(p−2) +O (1) +O (1)

=T(p-3) +O (1) +O (1) +O (1)

…

…

…

=O(p)

4) array\_inserting function is used to insert a new element into a young tableau

5) array\_inserting and arrayHeaping functions

6)finding\_element function determines whether a given number is stored in a given m x n young tableau