Name - Raveet Kumar
Roll No. - 1804310039
Branch - Computer Science and Engineering
College - BIET, Jhansi

QUESTION-

An m×n Young tableau is an m×n matrix such that the entries of each row are sorted order from left to right and entries of each column are in sorted order from top to bottom. Some entries of a Young tableau may be ∞, which we treat as nonexistent elements. Thus, a Young tableau can be used to hold r≤mn finite numbers. Show and write an algorithm how to insert a new element into a non-full m×n Young tableau in O(m+n) time. Hint: Y is a non-full 4×4 Young tableau.Y=[{2,5,8,9},{4,6,10,12},{6,7,13,14},{8,11,15,∞}] *

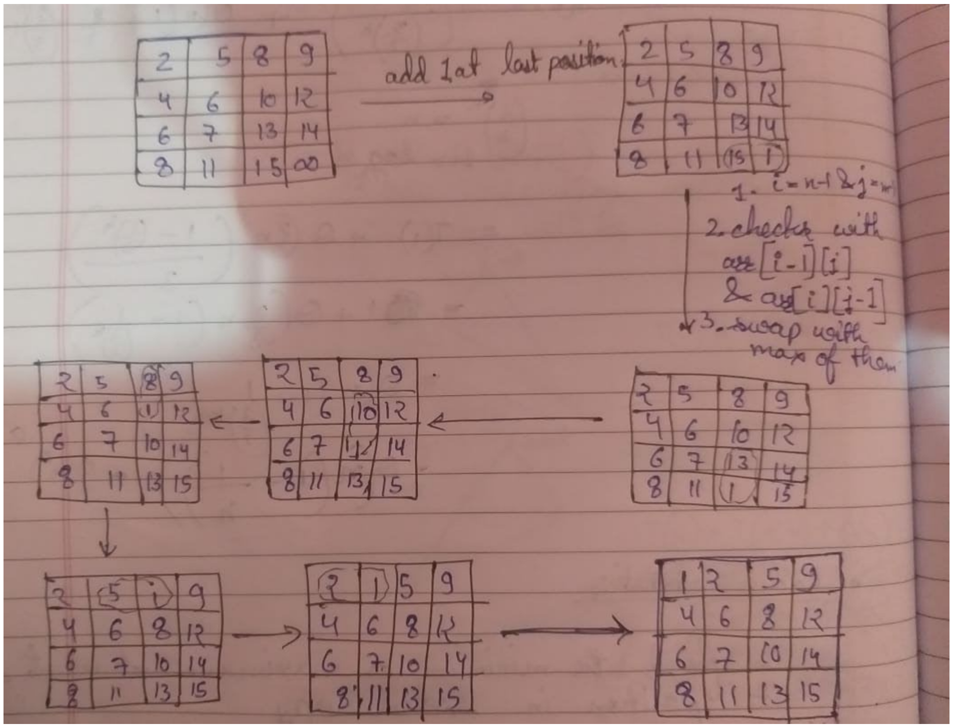
Array in starting:

2	5	8	9
4	6	10	12
6	7	13	14
8	11	15	infinity

Answer:

Insertion Algorithm:

For Ex- If we want to add 1 in the array.



```
Algorithm addTableau(arr[][],element,n,m){
      if( arr[n-1][m-1] != infinity){
          print(“tableau is filled”);
     }else{
        arr[n-1][m-1]=element;
        tableau(arr[][],n-1,m-1)
     }

}

Algorithm tableau(int[][] arr,int i,int j) {

       int largesti=i,largestj =j;

       if(  i-1>=0  && arr[i-1][j] > arr[largesti][largestj]) {
              largesti=i-1;
              largestj=j;
        }

       if(  j-1>=0  && arr[i][j-1] > arr[largesti][largestj]) {
              largesti=i;
              Largestj = j-1;
        }


       if(largesti !=i || largestj !=j) {
             int temp=arr[i][j];
             arr[i][j]=arr[largesti][largestj];
             arr[largesti][largestj]=temp;

             tableau(arr,largesti ,largestj);
       }else {
             return;
       }
    }
```

Final Answer:

1	2	5	9
4	6	8	12
6	7	10	14
8	11	13	15

2.

Similarly ,For deletion of we could delete from the start ,that will provide us the minimum element

```
Algorithm deleteTableauMin(arr[][],n,m){

      if(arr[0][0]==Integer.MAX_VALUE) {
             System.out.println("Tableau is empty");
       }else {
             arr[0][0]=Integer.MAX_VALUE;
             deleteMin(arr,0,0,m,n);
       }

}

Algorithm deleteMin(int[][] arr,int i,int j,m,n) {

       int smallesti=i,smallestj=j;

       if(  i+1< n  && arr[i+1][j] < arr[smallesti][smallestj]) {
              smallesti=i+1;
              smallestj=j;
        }

       if(  j+1< m  && arr[i][j+1] < arr[smallesti][smallestj]) {
              smallesti=i;
              smallestj=j+1;
        }

       if(smallesti !=i || smallestj !=j) {
             int temp=arr[i][j];
             arr[i][j]=arr[smallesti][smallestj];
             arr[smallesti][smallestj]=temp;

             deleteMin(arr,smallesti,smallestj);
       }else {
             return;
       }
    }
```

Final Answer:

2	5	8	9
4	6	10	12
6	7	13	14
8	11	15	infinity