

VNT

Test your Matrix class by using it to implement a VNT class.

We want a class called VNT (stands for “**Very Neat Table**”) which is going to handle an $m \times n$ matrix such that the entries of each row are in sorted order from left to right and the entries of each column are in sorted order from top to bottom. Some of the entries of a VNT may be `INT_MAX` , which we treat as nonexistent elements.

Thus, a VNT can be used to hold $r \leq mn$ integers

We want the class to be able to do the following things:

0. If A is a VNT object then `A[i][j]` is the i,j th element of the underlying 2 dimensional Matrix. So if A is empty then `A[0][0]=INT_MAX` and if A is full then `A[m-1][n-1] < INT_MAX`.

1. `VNT(int m, int n)` will create an $m \times n$ VNT object.

`VNT A(5,7);`

2. `A.add(25)` will add 25 to a non-full VNT

3. `A.getMin();` will extract the smallest element for A and leave A a VNT.

4. `A.sort(int k[], int size)` sort the $n \times n$ numbers in k using A and not calling any sort routine as a subroutine.

5. `A.find(int i)` will return true if i is in A and false otherwise.

The Constraints

- function `add()` should work in time proportional to $m+n$.
- function `getMin()` should work in time proportional to $m+n$.
- function `sort()` should work in time proportional to n^3 .
- function `find()` should work in time proportional to $m+n$.