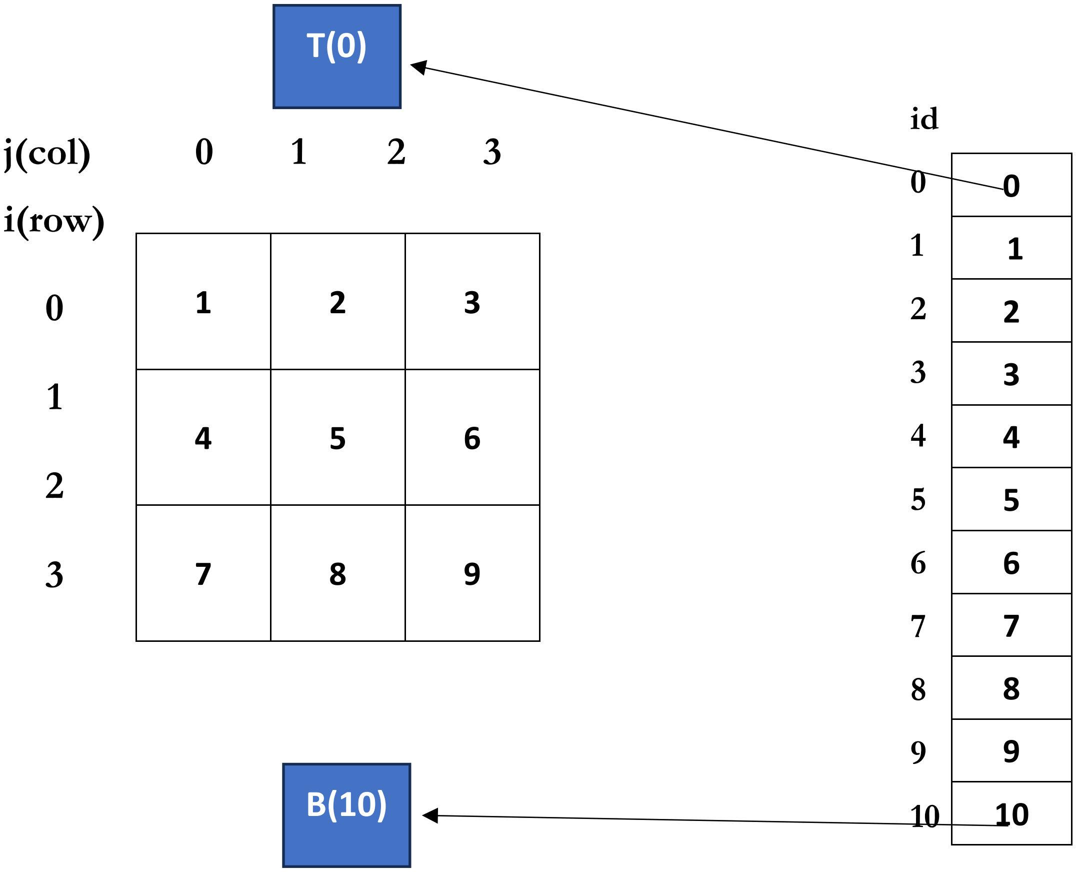
Wittenberg UNIVERSITY

Spring 2024

Course Title: Sequential and parallel Algorithm
Course Number & Section: COMP 275-01
Group Programming Project Assignment #1:
(Percolation Pseudocode)



1. Grid (2-D array: N x N) is used to capture(simulate) the concept of:
    - Open site
    - Block site
    - Full Site
2. The 1-D array is the one we used in implementing quick union and find algorithm for solving connectivity problems.
3. Grid Box number to 1-D array box number Translation Formula:
   Box Number = box-size(N) *i(row)+ [j(col) +1]

Pseudocode for Percolation API

1. **Class constants**
   - TOP = 0
2. **Class variables**
   - BOTTOM=10
   - Declare a boolen type 2-D grid Box array.
   - Declare an Open site counter.
   - Declare the Quick union and find Object.

3. **Constructor (int n){}**
   - Throw exception if n<0
   - Set value of BOTTOM = size of grid box + 1
   - Create the Quick union and find Object and the pass argument should be the size of grid box + 2
   - Create the boolen type 2-D grid Box array.
   - Set number of open site counter to 0
4. **Open(i,j) method:**
   - Before opening a site perform boundary check on i and j to ensure their value falls between 0 and N and if they are not, throw out-of-boundary exception.
   - Open the site i and j by setting it true.
   - If the open site (grid box at i & j) is on the top-most row of the grid, then union it with TOP
   - If the open site (grid box at i & j) is on the bottom-most row of the grid, then union it with BOTTOM
   - If the open site is neither the on top-most nor bottom-most row, check if the neighboring sites are opened and union it with those.
5. **isOpen(i, j) method:**
   - Just return the grid box value for i and i
6. **numberOfOpenSites() method:**
   - just return the number
7. **isFull(i,j) method:**
   - perform boundary check on i and j to ensure their value falls between 0 and N, and if they are not, throw out-of-boundary exception.
   - Using the union find method check if grid box (i,j) is connected to the any top-most box or any box that has some sort of connection to them.
   - Return a Boolean true if there is a form of connection otherwise false.
8. **Percolate () method:**

- Use the quick find method to check if top and bottom is connected then return true otherwise false.

Pseudocode for PercolationStats API

1. **Class constants**
   - CONFIDENCE = 1.96
2. **Class variables**
   - Declare a 1-D fraction array.
   - Declare number of open site
3. **Constructor (n, t){}**
   - Create fraction array of size equal to the value of t
   - Create a nested loop, the outer loop for n cycle and the inner loop for t cycle.
   - Create a percolate object between the outer loop and the inner loop.
   - Also, between the outer loop and the inner loop, set number of open sites to zero
   - The inner loop will stop only when the percolate object is true.
   - The inner loop will consist of:
     - i, which is a random number generated from a random number generator API(StdRandom) provided as aiding resource.
     - j, same as i
     - use i and i with the open() method in the percolation to open sites. So, you check first if a site at i and j is opened if not open it.
     - Increase number of open sites by 1
   - Following the end of the inner loop:
   - Compute the fraction of open sites with respect to the total number of sites.
   - Add the computed fraction of open sites to the fraction array.
4. **mean() method:**
   - compute the mean of the fraction array using the StdStats API provided in the assignment.
   - Return the computed value.
5. **stdDev() method:**
   - compute the mean of the fraction array using the StdStats API provided in the assignment.
   - Return the computed value.
6. **ConfidenceLow() and ConfidenceHigh() methods:**
   - Use the formula given the assignment doc to compute them.
   - Return their respective results accordingly,
7. **psvm method():**
   - define the values n and t by passing it as an argument to your program main() method or hardcoding it

- create percolation object and pass n and t as argument to it.
- Print the value of:
    a. Confidence
    b. Mean
    c. Standard deviation