

## CSC 226: Summer 2019: Lab 6

**Problem:** *Generate an spanning tree by randomly applying union operations on an  $N \times N$  grid and show the connections by animation.*

To solve this problem, we will use these classes from algs4.jar.

1. StdIn
2. StdOut
3. StdRandom
4. StdDraw
5. UF

You can find the documentation of each class at  
“[https://algs4.cs.princeton.edu/code/javadoc/edu/princeton/cs/algs4/\[classname\].html](https://algs4.cs.princeton.edu/code/javadoc/edu/princeton/cs/algs4/[classname].html)”.

We will now give an step-by-step algorithm, but you are welcome to try your own algorithm.

1. Download the template UFAnimation.java from connex.
2. In the main method, prompt the user to enter an integer using println() method from StdOut class.
3. Read the integer input by the user using readInt() method from StdIn and store it in a variable 'N'.
4. Call the setScale() method from StdDraw to set scales from -1 to N.
5. Clear the screen and set pencolor using the following lines of code:

```
StdDraw.clear(StdDraw.BLACK);  
StdDraw.setPenColor(StdDraw.WHITE);
```

6. Create an array of Point class, lets call it array and initialize each point with coordinates. This means you need two nested for loops, both going from 0 to  $N - 1$ , and call the Point constructor for each element of array.
7. Draw each point by calling the filledCircle() method from StdDraw. Use radius less than or equal to 0.05.

8. You can add a pause by calling the `pause()` method from `StdDraw`.
9. create an UF object:  
`UF unionFind = new UF(N*N);`
10. Run a while loop that should stop when  $N * N - 1$  connections have been made. Inside the while loop, generate two random numbers  $p$  and  $q$  in the range  $[0, N * N)$  by calling `uniform()` method from `StdRandom` class. Call `find()` for both  $p, q$ . If they are not connected, call `union()` for  $p, q$ . Show the newly formed connection by calling the `StdDraw.line()` method. You can give a pause for 300 milliseconds after drawing each line.
11. To run your code, download `algs4.jar` in the same folder as your java file and run the following commands on command prompt.

```
javac -cp ;algs4.jar UFAAnimation.java java -cp  
;algs4.jar UFAAnimation
```

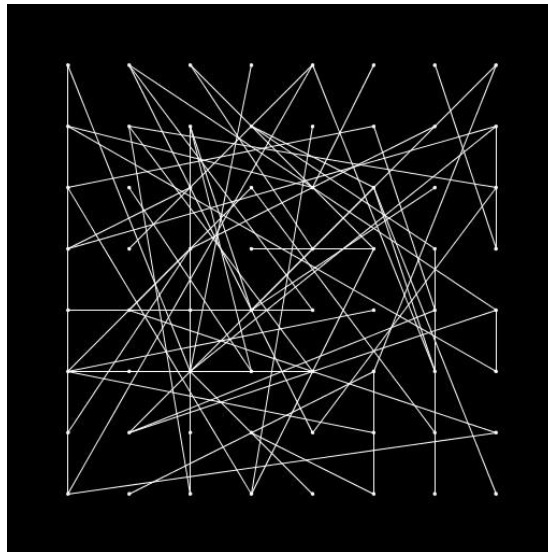


Figure 1: random MST on a  $8 \times 8$  grid.

The above algorithm will give you an MST where connections are made between any two points on the grid (see Figure 1). Now try to think of a way to modify your code so that each edge in the MST has length exactly 1 as shown in Figure 2.

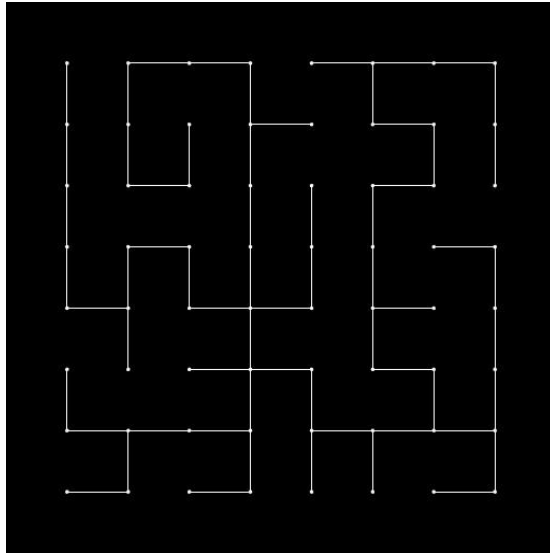


Figure 2: random MST on a  $8 \times 8$  grid where each edge of the MST has length 1.