You're given PercolationStats.java which performs benchmarks for an **IPercolate** object using grid sizes of 100, 200, 400, 800, 1600, and 3200. If you don't change the value of the public **RANDOM_SEED** variable you should ***see the same mean values*** of the Percolation threshold shown below. Your timing values may vary, but for all implementations using 20 trials you should see the same mean and standard deviation values. A sample run is provided below from running PercolationDFSFast on a head UTA's computer.

```
simulation data for 20 trials
grid    mean   stddev total time
100     0.593  0.014  0.160
200     0.591  0.010  0.194
400     0.590  0.006  1.249
800     0.594  0.004  6.003
Exception in thread "main" java.lang.StackOverflowError
        at PercolationDFS.dfs(PercolationDFS.java:109)
        at PercolationDFS.dfs(PercolationDFS.java:109)
        at PercolationDFS.dfs(PercolationDFS.java:109)
        at PercolationDFS.dfs(PercolationDFS.java:110)
        at PercolationDFS.dfs(PercolationDFS.java:109)
        [... rest truncated]
```

Copy/paste the results for each **IPercolate** object (**PercolationDFSFast**, **PercolationBFS**, **PercolationUF**) in your PDF document.

So first copy/paste data for the grid sizes shown above for all three IPercolate classes you implement for this project. Then answer these questions using data from **PercolationUF** with **QuickUWPC**.

1. How does doubling the grid size affect running time (keeping # trials fixed)
2. How does doubling the number of trials affect running time.
3. Estimate the largest grid size you can run in 24 hours with 20 trials. Explain your reasoning.

***After completing the analysis questions you should submit your answers as a PDF to the P6-analysis project on Gradescope.***

**PercolationDFSFast**

simulation data for 20 trials

| grid | mean | stddev | total time |
|---|---|---|---|
| 100 | 0.593 | 0.014 | 0.140 |
| 200 | 0.591 | 0.010 | 0.203 |
| 400 | 0.590 | 0.006 | 1.384 |
| 800 | 0.594 | 0.004 | 4.934 |

Exception in thread "main" java.lang.StackOverflowError

**PercolationBFS**

simulation data for 20 trials

| grid | mean | stddev | total time |
|---|---|---|---|
| 100 | 0.593 | 0.014 | 0.121 |
| 200 | 0.591 | 0.010 | 0.196 |
| 400 | 0.590 | 0.006 | 1.028 |
| 800 | 0.594 | 0.004 | 4.728 |
| 1600 | 0.592 | 0.002 | 28.858 |
| 3200 | 0.593 | 0.001 | 197.083 |

**PercolationUF**

simulation data for 10 trials

| grid | mean | stddev | total time |
|---|---|---|---|
| 100 | 0.593 | 0.019 | 0.143 |
| 200 | 0.596 | 0.006 | 0.165 |
| 400 | 0.592 | 0.006 | 0.404 |
| 800 | 0.592 | 0.003 | 2.240 |
| 1600 | 0.594 | 0.002 | 9.294 |
| 3200 | 0.593 | 0.001 | 62.374 |

simulation data for 20 trials

| grid | mean | stddev | total time |
|---|---|---|---|
| 100 | 0.593 | 0.014 | 0.107 |
| 200 | 0.591 | 0.010 | 0.111 |
| 400 | 0.590 | 0.006 | 0.746 |
| 800 | 0.594 | 0.004 | 3.753 |
| 1600 | 0.592 | 0.002 | 16.727 |
| 3200 | 0.593 | 0.001 | 83.909 |

1. **How does doubling the grid size affect running time (keeping # trials fixed)**

When the grid size increases from 400 to 800, running time increases from 0.746 to 3.753 approximately by a factor of 5; when the grid size increases from 800 to 1600, running time increases from 3.753 to 16.727 approximately by a factor of 4; when the grid size increases from 1600 to 3200, running time increases from 16.727 to 83.909 approximately by a factor of 5. Therefore, doubling the grid size approximately increases running time by a factor of 5.

2. **How does doubling the number of trials affect running time.**

When the grid size is 400, running time increases from 0.404 to 0.746 by a factor of 1.8 as the number of trials doubles; when the grid size is 800, running time increases from 2.240 to 3.753 by a factor of 1.7 as the number of trials doubles; when the grid size is 1600, running time increases from 9.294 to 16.727 by a factor of 1.8 as the number of trials doubles; when the grid size is 3200, running time increases from 62.374 to 83.909 by a factor of 1.3 as the number of trials doubles. On average, doubling the number of trials increases running time by a factor of 1.7. x

3. **Estimate the largest grid size you can run in 24 hours with 20 trials. Explain your reasoning.**

Running time for a grid of size 3200 is around 85s.
24h = 24*60*60 = 86400s
85*(5^n) = 86400
n = 4.3
3200*(4.3^2) = 59230
The largest grid size that can be run in 24h with 20 trials is around 59230.