Percolation Analysis

1. How does doubling the grid-size, N, affect the running time?
   Using 100 test counts, doubling the grid-size from 50 to 100 elements had a time change factor of 10 for PercolationDFS, 2 for PercolationDFSFast, 15 for PercolationUF with QuickFind, and 2 for PercolationDFSFast with Quick UWPC.

2. How does doubling the number of experiments, T, performed affect the running time?
   Using a grid-size of 100, doubling the number of experiments from 100 to 200 tests had a time change factor of 2 for PercolationDFS, 1.5 for PercolationDFSFast, 2 for PercolationUF with QuickFind, and 1.5 for PercolationDFSFast with Quick UWPC. Based on the code they all should be 2 because T changes the run time linearly.

3. Try to provide a formula for the running time in terms of N and T, use big-Oh.
   Run time for each of the classes increases linearly with T, because it only influences the amount of times the program is run. PercolationDFS's runtime is based on N^2 because the class has to go through the grid looking for connections twice per run so it has a big-Oh of TO(N^2). Percolation DFSFast's runtime is based on N because the class only loops through opening once per run so it has a big-Oh of TO(N). PercolationUF with Quickfind has a runtime based on N^2 because initializing the array loops N amount of times and calling union also has a runtime of N so it has a big-Oh of TO(N^2). PercolationUF with OuickUWPC has a runtime based on Nlog(N) because initializing the arrays loops N amount of times while union, find, and connected together take log(N) time, because finding roots becomes quicker as there becomes less components with more sites so it has a big-Oh of TO(Nlog(N).

4. Estimate the largest grid-size you can run in a day for 100 trials (assume time is the only limit here, not memory).
   For PercolationDFS the polynomial line of best fit equation based on the data was Time = $0.0032N^2 - 0.4073N + 11.191$. Estimating a grid-size for a run time of 24 hours ended up being 5,260 for an N value. For PercolationDFSFast the polynomial line of best fit equation based on the data was Time = 0.0078N - 0.5841. Estimating a grid-size for a run time of 24 hours ended up being 1.1077×10^7 for an N value. For PercolationUF using QuickFind the polynomial line of best fit equation based on the data was Time = $0.0036N^2 - 0.5171N + 15.394$. Estimating a grid-size for a run time of 24 hours ended up being 4,971 for an N value. For PercolationUF using QuickUWPC the polynomial line of best fit equation based on the data was Time = 0.0436N - 10.234. Estimating a grid-size for a run time of 24 hours ended up being 1.98189 x^6 an N value.

5. Give estimate for how much memory is used in terms of N, the grid-size. Provide your estimate in bytes and use four bytes for an int, one byte for a boolean, and eight bytes for a double. For example, an array of N integers uses 4N bytes in this model, there's no overhead for the array other than storing the integer values.
   PercolationDFS uses a NxN grid of integer values and an integer value to store myOpenCount so an estimate of storage is $4N^2+4$ Bytes. PercolationDFSFast stores the

same values as PercolationDFS so an estimate of storage is also $4N^2+4$. PercolationUF stores 10 integer values and N^2 boolean values in the grid and 2 more boolean values. QuickFind stores an array of N integers and 2 more integers. An estimate for storage of PercolationUF running QuickFind is $N^2 + 4N + 50$ Bytes. QuickUWPC stores 2 integer arrays of N values and 6 more integers. An estimate for storage of PercolaitionUF running QuickUWPC is $N^2+8N+66$ Bytes.

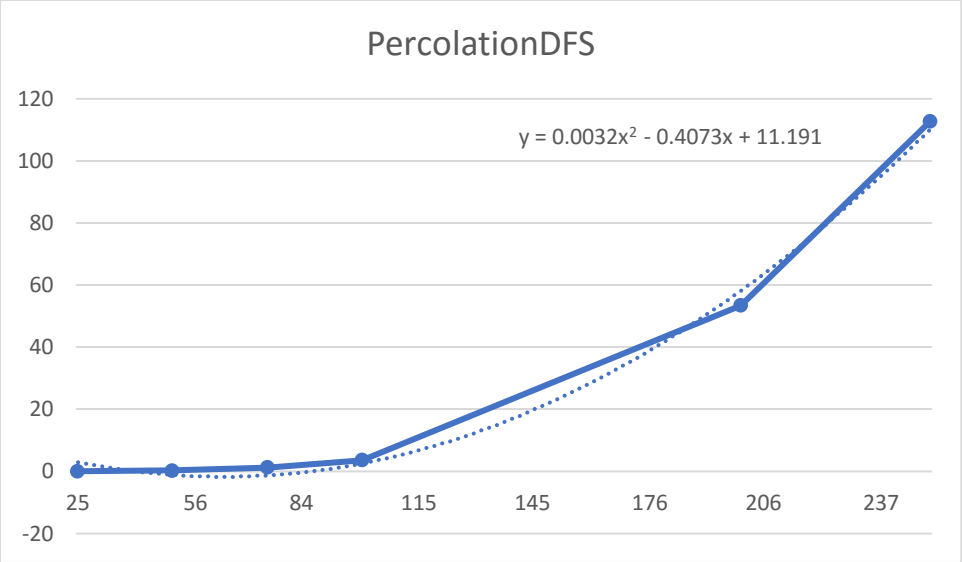
Data

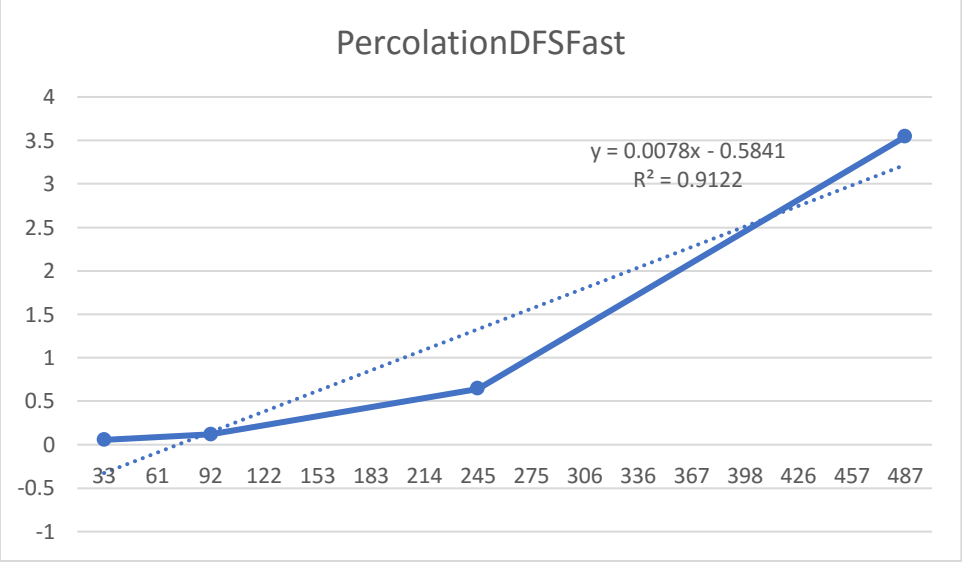
PercolationDFS			
Grid Size	Tests Count	Mean	Time
25	100	0.5933	0.0585
50	100	0.5918	0.3353
75	100	0.5931	1.3127
100	100	0.5944	3.5905
200	100	0.5931	53.4421
250	100	0.5932	112.6951
100	200	0.5928	7.4137

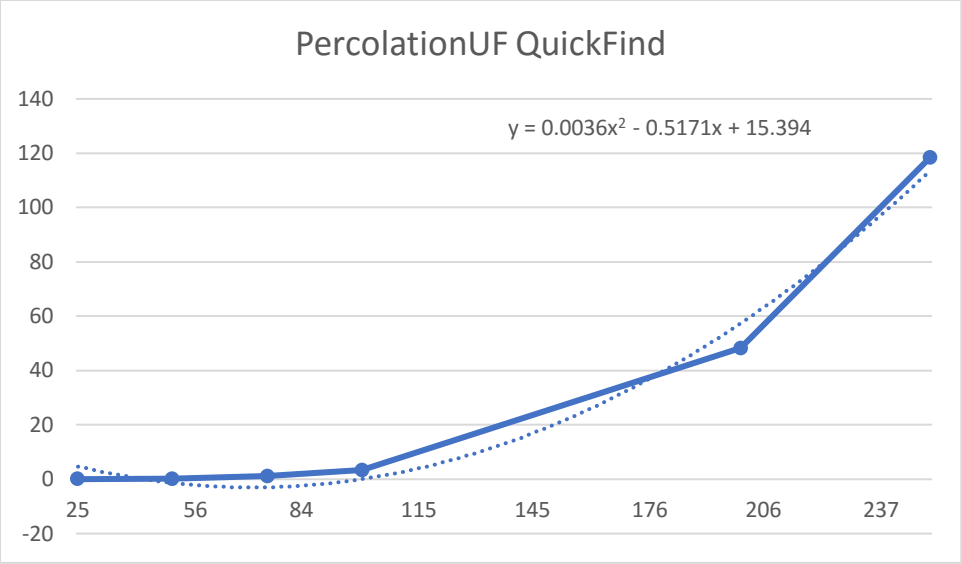
PercolationDFSFast			
Grid Size	Tests Count	Mean	Time
50	100	0.5918	0.0552
100	100	0.5944	0.117
250	100	0.5932	0.642
500	100	0.5923	3.5426
1000	100	error	error
2000	100	error	error
100	200	0.5928	0.1798

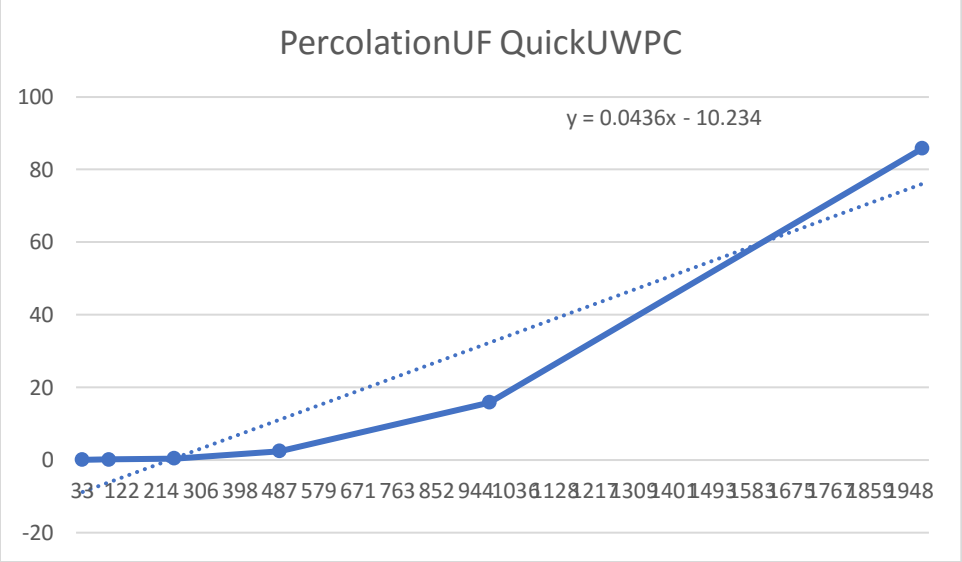
PercolationUF QuickFind			
Grid Size	Tests Count	Mean	Time
25	100	0.5933	0.0349
50	100	0.5918	0.2173
75	100	0.5931	1.0786
100	100	0.5944	3.4039
200	100	0.5931	48.2563
250	100	0.5932	118.4322
100	200	0.5928	6.6549

PercolationUF QuickUWPC			
Grid Size	Tests Count	Mean	Time
50	100	0.5918	0.0585
100	100	0.5944	0.1311
250	100	0.5932	0.4722
500	100	0.5923	2.456
1000	100	0.5929	15.8642
2000	100	0.5925	85.8345
100	200	0.5928	0.1787






PercolationUF QuickFind
y = 0.0036x² - 0.5171x + 15.394
140
120
100
80
60
40
20
0
-20
25
56
84
115
145
176
206
237


PercolationUF QuickUWPC
y = 0.0436x - 10.234
100
80
60
40
20
0
-20
33
122
214
306
398
487
579
671
763
852
944
1948