Kenneth Marenco
kem78

Copy/Paste from running PercolationStats with these grid sizes:
grid sizes of 100, 200, 400, 800, 1600, and 3200
and using 20 trials

PercolationDFSFast

simulation data for 20 trials

grid	mean	stddev	total time
100	0.593	0.014	0.098
200	0.591	0.010	0.143
400	0.590	0.006	0.939
800	0.594	0.004	6.753

Exception in thread "main" java.lang.StackOverflowError
at PercolationDFS.dfs(PercolationDFS.java:108)
at PercolationDFS.dfs(PercolationDFS.java:108)
at PercolationDFS.dfs(PercolationDFS.java:108)
at PercolationDFS.dfs(PercolationDFS.java:108)

PercolationBF

simulation data for 20 trials

grid	mean	stddev	total time
100	0.593	0.014	0.101
200	0.591	0.010	0.163
400	0.590	0.006	0.964
800	0.594	0.004	6.979
1600	0.592	0.002	29.423
3200	0.593	0.001	187.221

PercolationUF with QuickUWPC

simulation data for 20 trials

grid	mean	stddev	total time
100	0.593	0.014	0.087
200	0.591	0.010	0.129
400	0.590	0.006	0.765
800	0.594	0.004	4.586
1600	0.592	0.002	20.155
3200	0.593	0.001	100.109

Answer these questions for PercolateUF with a QuickUWPC union-find object

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

By the simulation data, it can be seen that the runtime over doubles when we double the grid size. Using a graphing calculator with the data collected, I used quadratic regression to find a curve that fit the data. This led to the fit with the equation
$y=1.15557*10^{(-5)}x^2 + -.00606x + 1.047368$
This gave an $R^2 = .999751$ which supports that this program has a runtime of $O(N^2)$.

How does doubling the number of trials affect running time.

When I changed the # of trials to 40 (aka doubling the trials) the runtime for each grid just about doubled. The simulation time is below. The makes sense since the class is working on the same size grid just doing more examples of it which would be a linear multiplicative operation. The average runtime for a singular trial is about the same as can be seen from the simulation data.

simulation data for 40 trials

grid	mean	stddev	total time
100	0.594	0.015	0.125
200	0.591	0.009	0.256
400	0.591	0.005	1.627
800	0.593	0.004	9.255
1600	0.593	0.002	39.545
3200	0.593	0.001	211.524

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

Using the equation I found in question 1 (y=1.15557*10^(-5)x^2 + -.00606x + 1.047368), I was able to estimate the largest grid that could be solved with the constraints above was about 669754 with an order of magnitude 100,000.