251 project 1 analysis

Jackson Didat

L07

Data for WeightedQuickUnionUF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10 | 25 | 50 | 100 | 250 | 500 |
| mean threshold | .61 | .59579 | .59916 | .59416 | .59076 | .59460 |
| std dev | 0.04599 | 0.02860 | 0.02372 | 0.01356 | 0.00605 | 0.00521 |
| time (in secs) | 0.007 | 0.02200 | 0.03000 | 0.08900 | 0.59500 | 2.95099 |
| mean time | 2.33333E-4 | 7.33333E-4 | 0.00100 | 0.00297 | 0.01983 | 0.09837 |
| stddev time | 5.04007E-4 | 0.00139 | 0.00158 | 0.00254 | 0.01058 | 0.09143 |

Data for QuickUnionUF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10 | 25 | 50 | 100 | 250 | 500 |
| mean threshold | .613 | .59381 | .59776 | .59763 | .59016 | .59093 |
| std dev | 0.047051 | 0.02702 | 0.02092 | 0.01363 | 0.00803 | 0.00568 |
| time (in secs) | 0.008 | 0.01700 | 0.04300 | 0.13500 | 1.2 | 12.765 |
| mean time | 2.66667E-4 | 5.66667E-4 | 0.00143 | 0.00450 | 0.04 | 0.42550 |
| stddev time | 5.20830E-4 | 9.71431E-9 | 0.00152 | 0.00424 | 0.02142 | 0.33014 |

For the first half of the tests (10, 25, 50), the time complexity for both algorithms were very small and very close to each other. However, as N got much larger, the time complexity increased exponentially. It became clear that Weighted Union Find is a much faster and more efficient algorithm than Quick Union Find. For example, an N size of 50 had a time complexity of 0.03000 seconds for Weighted, while Quick has a time complexity of 0.04300 seconds. The difference between the two times is extremely small. However, the time complexity for Weighted with N being 500 has a time complexity of 2.95099 seconds while Quick has a time complexity of 12.765 for N being 500. This is a huge difference in runtimes. And as N increases, the gap between the time complexities of the two algorithms increases at an exponential rate.

Quick Union Find has a slower time complexity than Weighted Union Find because Weighted Union Find uses if statements to check which index that is send to the union is the largest. Doing this has the smaller index used as the parent array first and it is set equals the larger index. This dramatically increases the time complexity for Quick Union Find.

The behavior of \*p across all tests for N is almost always between approx .585 and .599. This means that on average, approx. 3/5 of the grid is opened when the system reaches percolation. This is because \*p is a **ratio** of the number of grid opens it took for the system to percolates and the grid size. The ratio should stay the same since the number of opens and \*p should always be in this range because it is statistically what it should be.



