

Name: Zhaoxi Zhang

NetID: zz115

Hours Spent: 5 to achieve the desired function and another 7 to make gradescope happy and finish the analysis, 12 in total.

Consulted With: NONE

Resources Used: Original Lecture video of Union Find from Princeton

Impressions: Algorithm is not complicate, but the assignment description is very vague. The function of many method is not specified clearly enough. For example, open (which actually calls dfs) and isFull(in UF it actually also does the marking).

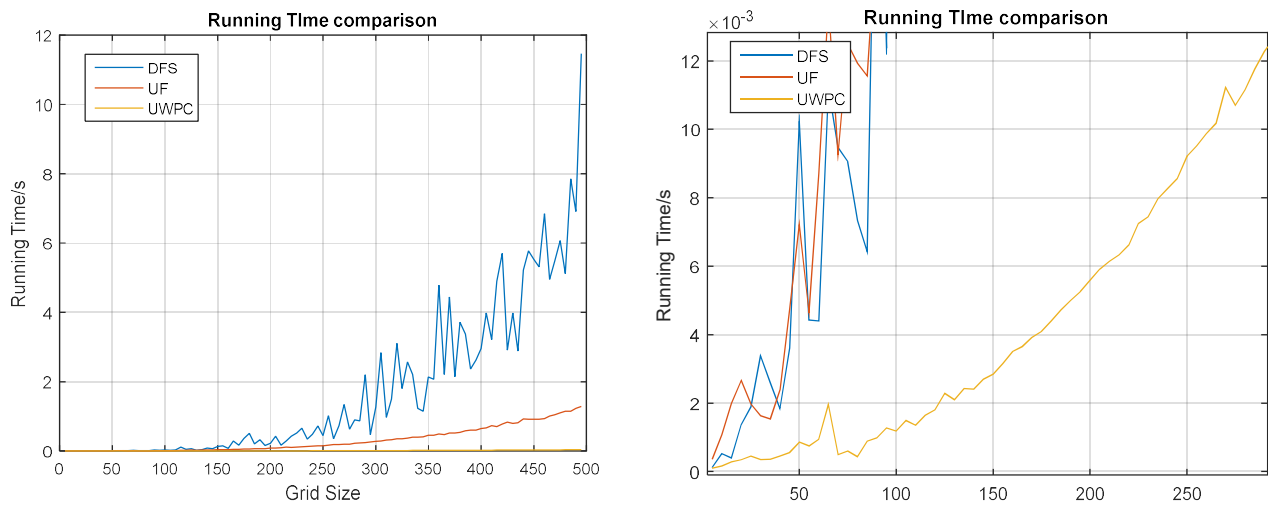


Figure 1 Running time comparison among DFS, UF, and UWPC

Problem 1: How does doubling N affect the running time?

DFS also has a N^3 running time growth with N in average. In the worst case, DFS will run on all the cells on the first row (N) and reach almost every cell except the bottom (N^2), and therefore

$$N \cdot N^2 = N^3$$

In total:

$$RunTime_{2n} = 8 \cdot RunTime_n$$

UF has a N^3 running time growth with N, as for each new open cell (N), there will be a linear scan through all cells (N), and considering the union find operations

$$RunTime_{2n} = 8 \cdot RunTime_n$$

UWPC in comparison, has an almost quadratic running time growth with N, as shown in the zoom plot in figure 1, but with a much smaller coefficient. Relative to the UF, the coefficient is much smaller, as no linear scan through all cells is required, but, there will still be a considerable number of union operations.

$$RunTime_{2n} \approx 4 \cdot RunTime_n$$

Problem 2: How does doubling T affect the running time?

T does not affect the running time for a single trial, as it does not affect the input size, grid size at all. T just increases the trial numbers and therefore it will cause a linear run time with T.

$$RunTime_T = 2 \cdot RunTime_{2T}$$

Problem 3: Measure running time (using calls to `System.currentTimeMillis`) of the three versions of your program (DFS, Quick Find, and weighted quick union with path compression).

For a single trial:

3A DFS: 0.001806793 s

3B Quick Find: 0.003479498 s

3C Weighted quick union with path compression: 5.64603E-4 s

The running time can be estimated with trial number using the relationship described in problem 2.

Problem 4: Give a formula (using Big-Oh notation) of the running time on your computer (in seconds) as a function of both N and T .

$$DFS: O(TN^3)$$

$$UF: O(TN^3)$$

$$UWPC: O(TN^2)$$

Problem 5: Estimate the largest experiment you could perform in one [minute, day, year] assuming your computer has enough memory.

As the UWPC has the best big O run time, the following estimation are based on UWPC.

5A 1 minute: 17343

5B 1 day: 587966

5C 1 year: $1.02512 \cdot 10^7$

Problem 6: Give a formula (using Big-Oh notation) that describes the amount of memory (in bytes) that your `IPercolate` implementation consumes as a function of N .

6A DFS: $O(N^2)$

6B Quick Find: $O(N^2)$

6C Weighted quick union with path compression: $O(N^2)$