Akash Bharadwaj Karthik (002787011)

Program Structures & Algorithms
Spring 2023 (Sec -8)
Assignment No. 4

Task

- (Part 1) *Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC.*

- (Part 2) *Using your implementation of UF_HWQUPC, develop a UF ("union-find") client that takes an integer value n from the command line to determine the number of "sites." Then generates random pairs of integers between 0 and n-1, calling connected () to determine if they are connected and union () if not. Loop until all sites are connected then print the number of connections generated. Package your program as a static method count () that takes n as the argument and returns the number of connections.*

- (Part 3) *Determine the relationship between the number of objects (*n*) and the number of pairs (*m*) generated to accomplish this (i.e., to reduce the number of components from* n *to 1). Justify your conclusion in terms of your observations and what you think might be going on.*

Relationship Conclusion:

The relationship between the number of sites (n) and the number of connections is:

$$\boldsymbol{Number\ of\ connections = n - 1}$$

The relationship between the number of objects (n) and the number of pairs (m) generated to reduce the number of components from n to 1 is found to be:

$$\boldsymbol{m = \frac{1}{2} n \ln n}$$

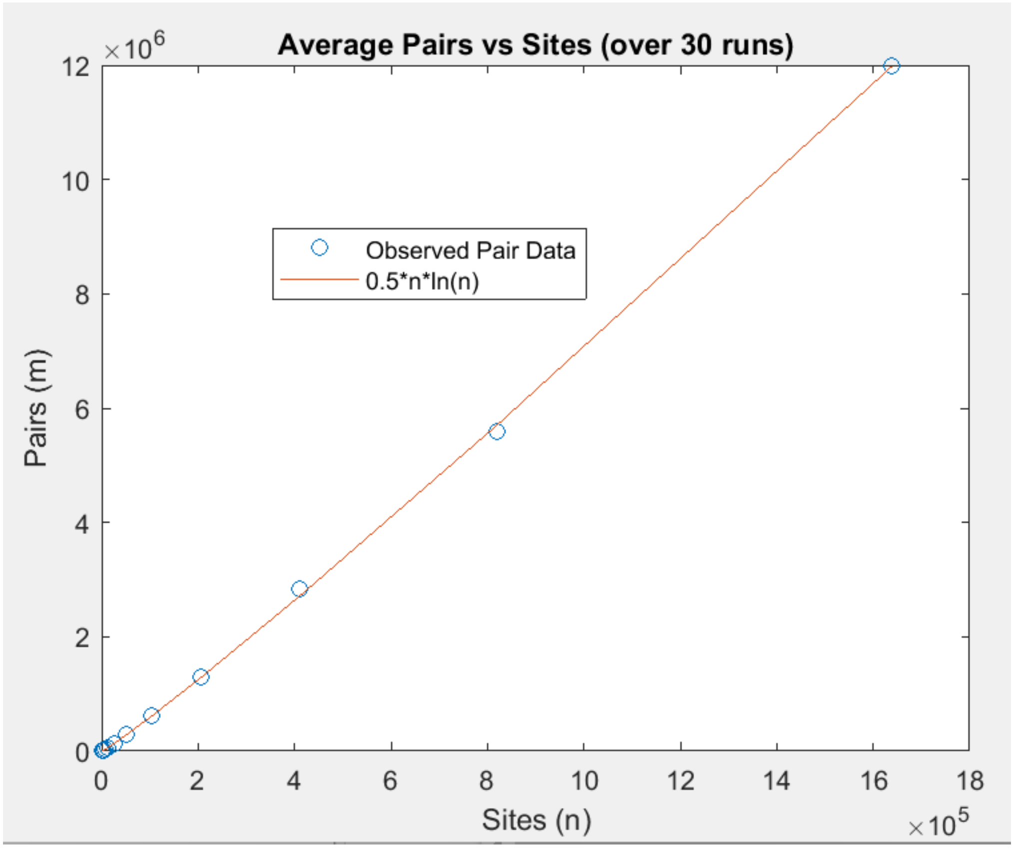
Evidence to support the conclusion:

A table of values recording sites - ‘N’, number of runs - ‘runs’, Connections – ‘connections’, Pairs – ‘m’, and hypothesized model - $\frac{1}{2} n \ln n$ is generated.

Values of ‘N’ are doubled, and increment as follows: 100,200,400, …,1638400.

Sites (n)	Average Connections Made	Average no of Pairs (m)	0.5*n*ln(n)
100	99	253.27	230.2585093
200	199	579.13	529.8317367
400	299	1420.4	1198.292909
800	399	2878.07	2673.844691
1600	499	6500.07	5902.207127
3200	599	13873.67	12913.44974
6400	699	29782.53	28044.97046
12800	799	69229.67	60526.08288
25600	899	137495.6	129924.4497
51200	999	300831	277593.4672
102400	1099	616591	590676.07
204800	1199	1302929.33	1252330.411
409600	1299	2844223.47	2646617.365
819200	1399	5583123.6	5577147.815
1638400	1499	11995914.73	11722121.8

Fitting observed data (blue circles) to model function (0.5*n*ln(n)) on MATLAB:



Analysis:

The number of ***connections is always n-1***. This makes sense because there would have to be at least n – 1 union operations to connect the n initial sites, and there can be at most n – 1 union operations, since more than n – 1 union operations would imply more than n connected components existed; but this is not possible since the maximum number of connected components is n at the start of the algorithm with 1 element in each component. Hence, we have shown that:
(n – 1) <= number of connections <= (n – 1). Hence connections = n – 1.

Minimum number of pair generation is expected to be equal to the number of connections. This is the optimistic case, since we are assuming we will always pick pairs leading to a union, and there can be **n-1** such unions until all objects are connected.

But in the ***average case***, we could end up picking pairs already in connected components, and hence the total number of pair generations is expected to be **more than n-1.**

If we assume the number of repetitions in generation of pairs scales as a harmonic series with a factor of n, we get:

Expected number of pair generations:

$$
\begin{aligned}
&= \boldsymbol{n} + \frac{\boldsymbol{n}}{\mathbf{2}} + \frac{\boldsymbol{n}}{\mathbf{3}} + \frac{\boldsymbol{n}}{\mathbf{4}} + \cdots \\
&= \boldsymbol{n}\left(\mathbf{1} + \frac{\mathbf{1}}{\mathbf{2}} + \frac{\mathbf{1}}{\mathbf{3}} + \frac{\mathbf{1}}{\mathbf{4}} + \cdots\right) \\
&\leq \boldsymbol{n} \ln \boldsymbol{n}
\end{aligned}
$$

Thus, our model function of **m =** $\frac{1}{2}\boldsymbol{n}\ln\boldsymbol{n}$ is a reasonable estimate.

Console Output:

```
Run:    UF_Client_Driver ×    UF_HWQUPC_Test ×
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Number of objects (n): 100, Number of connections = 99.0, Number of pairs (m) :253.27
Number of objects (n): 200, Number of connections = 199.0, Number of pairs (m) :579.13
Number of objects (n): 400, Number of connections = 399.0, Number of pairs (m) :1420.40
Number of objects (n): 800, Number of connections = 799.0, Number of pairs (m) :2878.07
Number of objects (n): 1600, Number of connections = 1599.0, Number of pairs (m) :6500.07
Number of objects (n): 3200, Number of connections = 3199.0, Number of pairs (m) :13873.67
Number of objects (n): 6400, Number of connections = 6399.0, Number of pairs (m) :29782.53
Number of objects (n): 12800, Number of connections = 12799.0, Number of pairs (m) :69229.67
Number of objects (n): 25600, Number of connections = 25599.0, Number of pairs (m) :137495.60
Number of objects (n): 51200, Number of connections = 51199.0, Number of pairs (m) :300831.00
Number of objects (n): 102400, Number of connections = 102399.0, Number of pairs (m) :616591.00
Number of objects (n): 204800, Number of connections = 204799.0, Number of pairs (m) :1302929.33
Number of objects (n): 409600, Number of connections = 409599.0, Number of pairs (m) :2844223.47
Number of objects (n): 819200, Number of connections = 819199.0, Number of pairs (m) :5583123.60
Number of objects (n): 1638400, Number of connections = 1638399.0, Number of pairs (m) :11995914.73

Process finished with exit code 0
```

Unit Tests:

UF_HWQUPC_Test.