Xiyue Suo(001348347)

Program Structures & Algorithms

Fall 2021

Assignment No. 3

- **Task (List down the tasks performed in the Assignment)**
  - Step 1:

- (a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you have to do is to fill in the sections marked with // TO BE IMPLEMENTED ... // ...END IMPLEMENTATION.

  Code Location: /Users/suoxiyue/IdeaProjects/INFO6205/src/main/java/edu/neu/coe/info6205/union_find/UF_HWQUPC.java

- (b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).

  - Step 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; and a main() that takes n from the command line, calls count() and prints the returned value. If you prefer, you can create a main program that doesn't require any input and runs the experiment for a fixed set of n values. Show evidence of your run(s).

- Code Location: /Users/suoxiyue/IdeaProjects/INFO6205/src/main/java/edu/neu/coe/info6205/union_find/UF_Client.java

- Step 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: (For ex : z = a * b)**

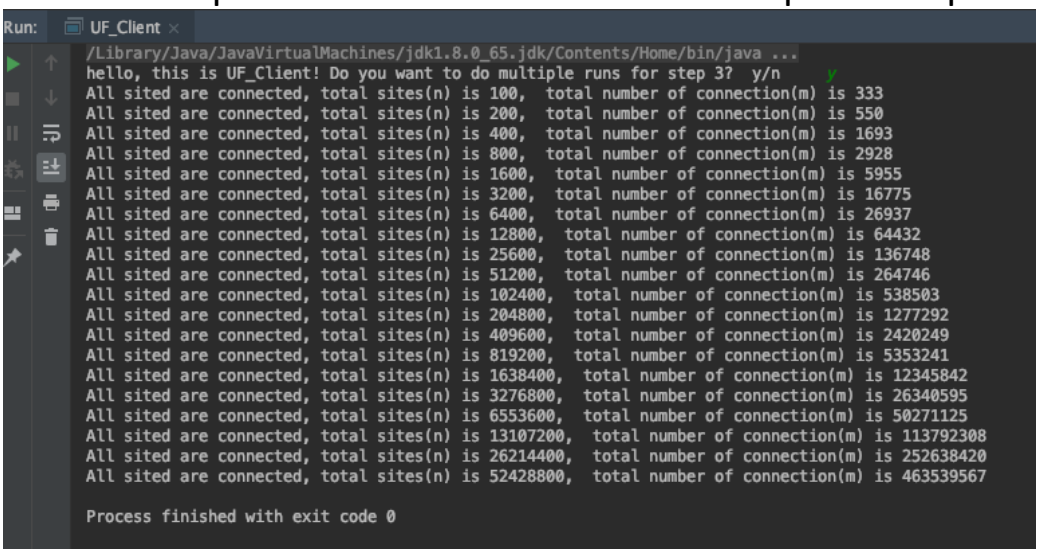
  - The relationship I got for the total number of objects (N) and the number of pairs (M) generated is:M = 1.1762*N*Log (N), where N is total sites, M is total generations.

    - I first found a linear equation with Excel:
    - M = 8.9841 * N – 650502, which is wrong. It has very big errors when n is smaller than 30000.
    - Then I tried M = N*Log (N), gives me -17.62% difference in average from actual data, the difference is almost constant with ration 0.15.
    - Thus, I tried M = 1.1762*N*Log (N), which results small offsets constantly, and gives a good relationship, the average difference from original M is around -3.1%.
    - Also I tried with logarithmic equation $\mathrm{M} = \frac{1}{2}NLnN$, this one give very close difference: -5.16%.

- **Evidence to support the conclusion:**

1. **Output (Snapshot of Code output in the terminal)**

Output of 20 runs of union find with path compression:

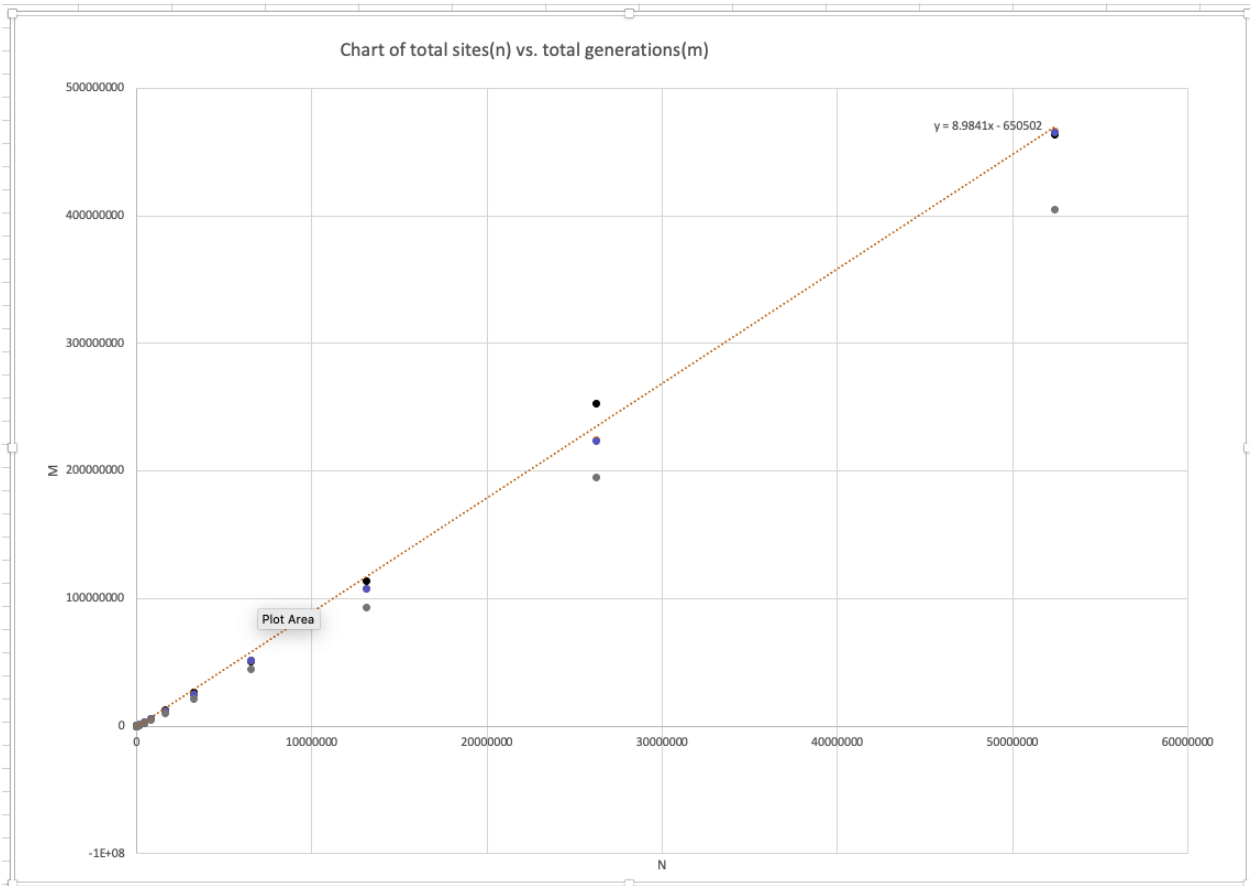


2. **Graphical Representation (Observations from experiments should be tabulated and analyzed by plotting graphs (usually in excel) to arrive on the relationship conclusion)**

   a. Table of Quick Union with Path Compression Data (n is sites, m is total generations)

	B	C	D	E	F	G	H	I	J	K
1	Quick Union With Path Compression Data Table (n is sites, m is total generations)									
2	N	M	M = 8.9841N - 650502	difference	1/2NLnN	diference	1.1762*N*logN	diference	N*logN	difference
3	100	333	-649604	too much	230	-0.3093	235	-0.2936	200	-0.3994
4	200	550	-648705	too much	530	-0.0364	541	-0.0158	460	-0.1633
5	400	1693	-646908	too much	1198	-0.2924	1224	-0.2769	1041	-0.3852
6	800	2928	-643315	too much	2674	-0.0867	2732	-0.0670	2322	-0.2068
7	1600	5955	-636127	too much	5902	-0.0089	6030	0.0126	5127	-0.1391
8	3200	16775	-621753	too much	12913	-0.2302	13193	-0.2135	11216	-0.3314
9	6400	26937	-593004	too much	28045	0.0411	28652	0.0637	24360	-0.0957
10	12800	64432	-535506	too much	60526	-0.0606	61836	-0.0403	52572	-0.1841
11	25600	136748	-420509	too much	129924	-0.0499	132735	-0.0293	112851	-0.1748
12	51200	264746	-190516	too much	277593	0.0485	283599	0.0712	241115	-0.0893
13	102400	538503	269470	too much	590676	0.0969	603455	0.1206	513055	-0.0473
14	204800	1277292	1189442	get better	1252330	-0.0195	1279424	0.0017	1087760	-0.1484
15	409600	2420249	3029385	get better	2646617	0.0935	2703875	0.1172	2298823	-0.0502
16	819200	5353241	6709273	get better	5577148	0.0418	5697806	0.0644	4844249	-0.0951
17	1638400	12345842	14069047	get better	11722122	-0.0505	11975722	-0.0300	10181706	-0.1753
18	3276800	26340595	28788597	get better	24579896	-0.0668	25111666	-0.0467	21349826	-0.1895
19	6553600	50271125	58227696	get better	51431097	0.0231	52543774	0.0452	44672483	-0.1114
20	13107200	113792308	117105894	get better	107404802	-0.0561	109728434	-0.0357	93290626	-0.1802
21	26214400	252638420	234862289	get better	223894824	-0.1138	228738640	-0.0946	194472573	-0.2302
22	52428800	463539567	470375080	get better	465960085	0.0052	476040823	0.0270	404727787	-0.1269
23					Average:	-0.0516	Average:	-0.0310	Average:	-0.1762
24						Good: -5.16%		Best: -3.1%		Ok: -17.62%

b. Graph of total sites N vs. total generations M



Black dots -- original data points.
Orange line -- linear trendline. (Huge difference)
Gray dots -- N*Log(N). (-17.62%)
Purple dots – 1.1762*N*Log(N). (-3.1%)
Green dots -- ½*N*Ln(N). (-5.16%)

From the table and chart, we can conclude that M = 1.1762*N*Log (N) fits best.

3. **Unit tests result:(Snapshot of successful unit test run**

All test in UF_HWQUPC_Test.java passed