Program Structures and Algorithms

Spring 2023

Assignment No : 3

Name: Srikanth Nandikonda

NUID: 002737724

Task:

Step 1: Implement height-weighted Quick Union with Path Compression and check that the unit tests for this class.

Step 2: 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).

Step 3: 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).

Relationship Conclusion:

Based on the value of m(number of pairs) by running the program 50 times for multiple values of n starting from 100 and doubling it till the value reaches 3276800, derived an relation between m and n

m is proportional to n*log(n)

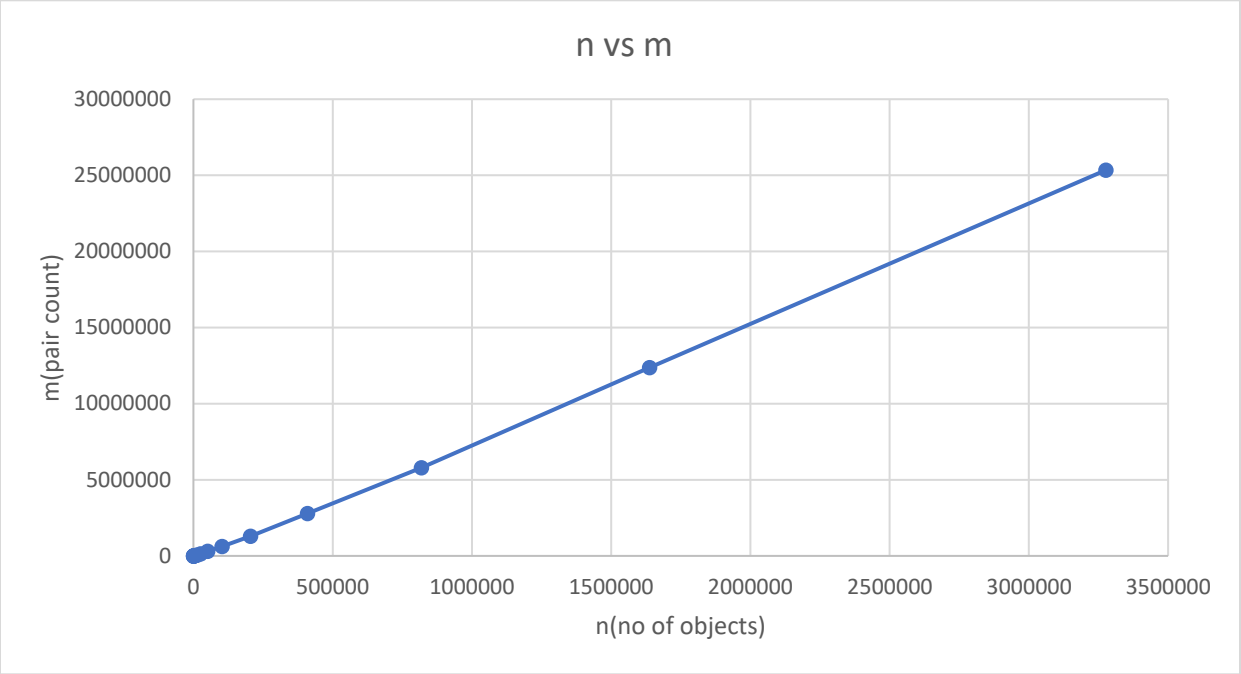
m is number of random pairs to make the number of connections from n to 1

n is the number of elements in the union find

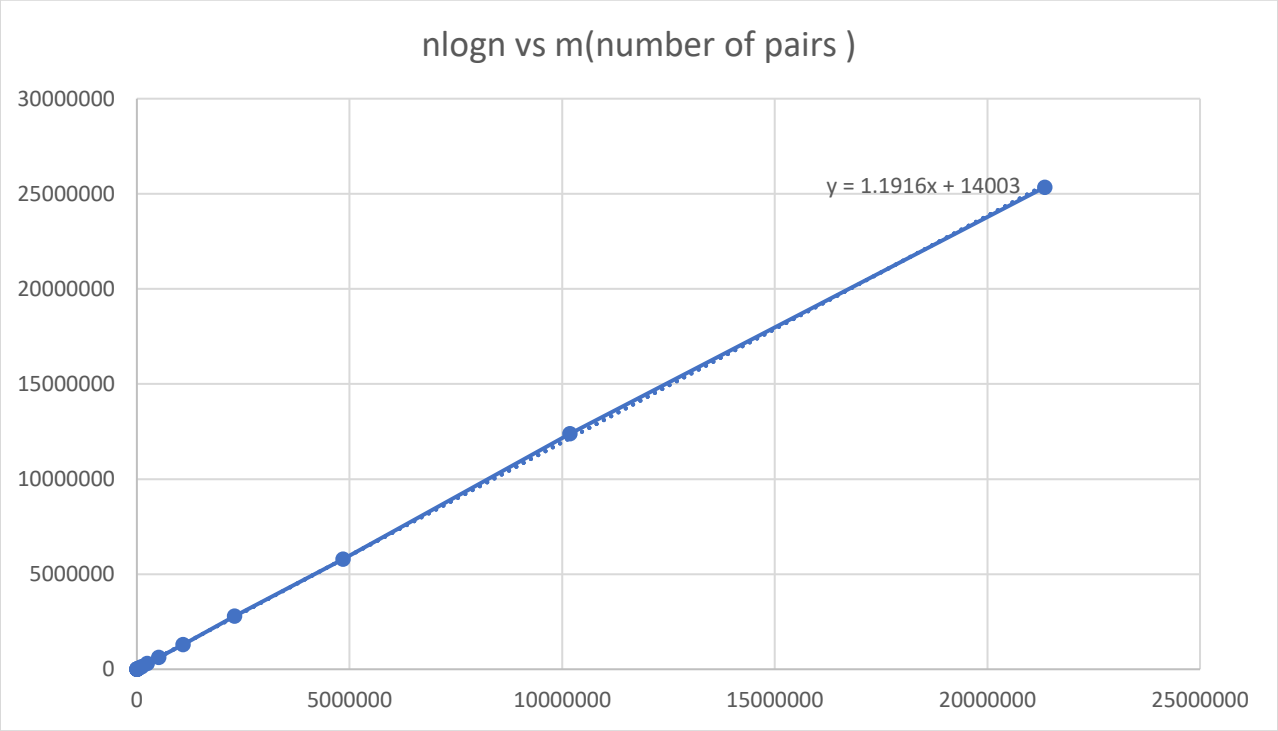
Evidence:

n(number of elements	nlogn(value)	m(value obtained from progra	Equation obtained from the graph
100	200	274	238.32
200	460.2059991	601	548.3814686
400	1040.823997	1352	1240.245874
800	2322.47199	2770	2767.457623
1600	5126.591972	6204	6108.846994
3200	11216.47993	13891	13365.55749
6400	24359.55183	29681	29026.84196
12800	52572.28761	64643	62645.13792
25600	112850.9431	137661	134473.1838
51200	241114.622	299784	287312.1836
102400	513054.7156	620829	611355.9991
204800	1087760.374	1289655	1296175.262
409600	2298822.635	2784870	2739277.051
819200	4844249.042	5783480	5772407.158
1638400	10181705.63	12373986	12132520.43
3276800	21349826.35	25345048	25440453.07

Graph of n (number of elements) vs m (number of connections to make it from n to 1)



Graph of n*log(n) vs m(number of pairs obtained from code)

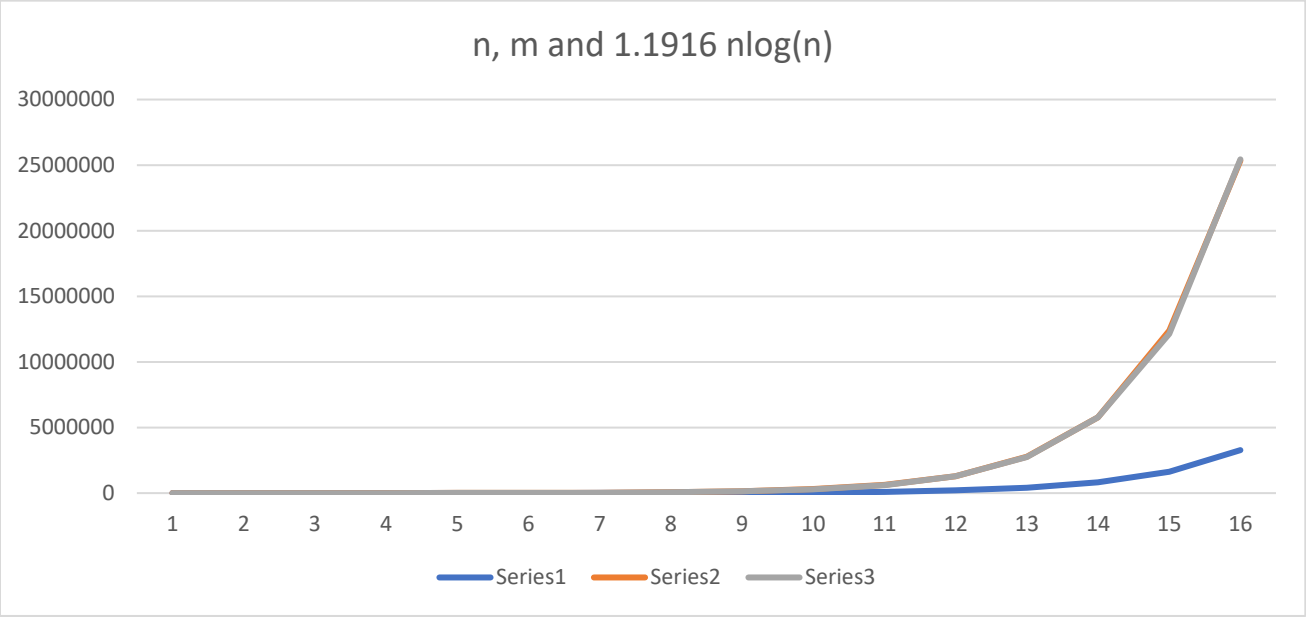


Using the trendline feature, obtain the linear equation

$$y = 1.1916x + 14003$$

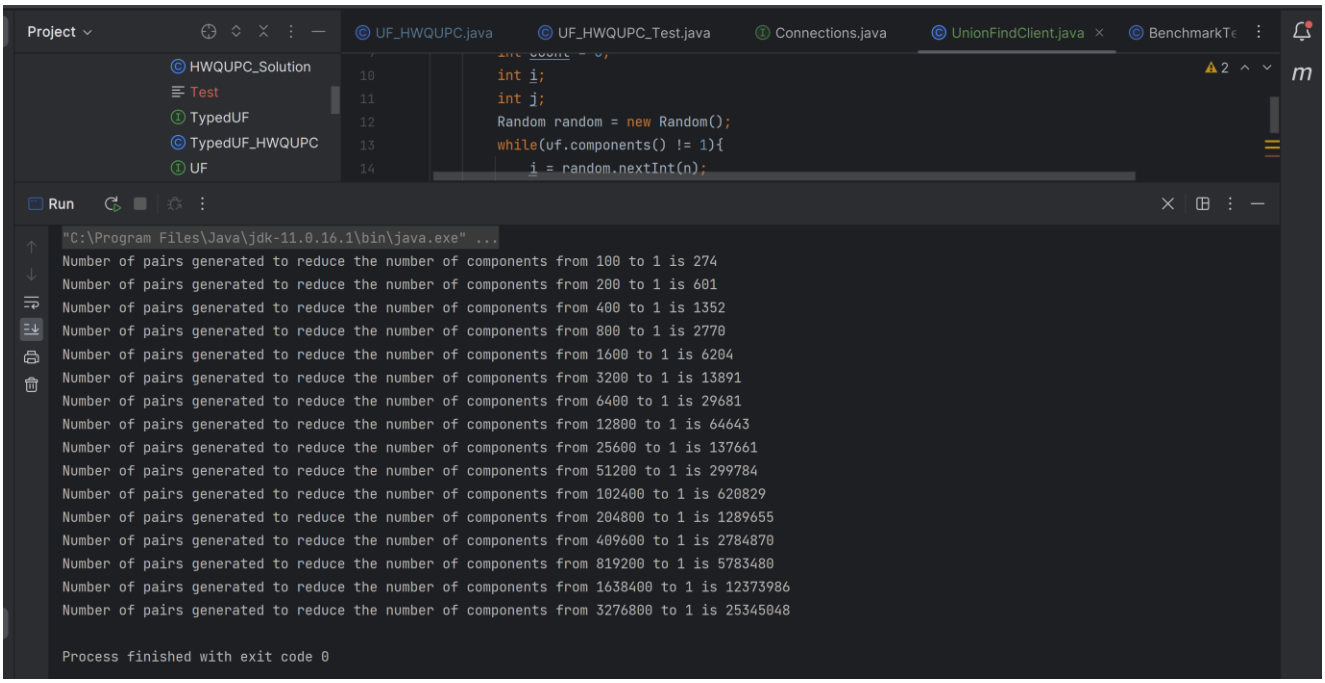
$$m = 1.1916 * n \log(n)$$

Graph three values on a single graph



Plot of m and 1.11916 n log(n) are very correlated, would be the best fit for out data points

Output Screenshot



Output

Number of pairs generated to reduce the number of components from 100 to 1 is 274

Number of pairs generated to reduce the number of components from 200 to 1 is 601

Number of pairs generated to reduce the number of components from 400 to 1 is 1352

Number of pairs generated to reduce the number of components from 800 to 1 is 2770

Number of pairs generated to reduce the number of components from 1600 to 1 is 6204

Number of pairs generated to reduce the number of components from 3200 to 1 is 13891

Number of pairs generated to reduce the number of components from 6400 to 1 is 29681

Number of pairs generated to reduce the number of components from 12800 to 1 is 64643

Number of pairs generated to reduce the number of components from 25600 to 1 is 137661

Number of pairs generated to reduce the number of components from 51200 to 1 is 299784

Number of pairs generated to reduce the number of components from 102400 to 1 is 620829

Number of pairs generated to reduce the number of components from 204800 to 1 is 1289655

Number of pairs generated to reduce the number of components from 409600 to 1 is 2784870

Number of pairs generated to reduce the number of components from 819200 to 1 is 5783480

Number of pairs generated to reduce the number of components from 1638400 to 1 is 12373986

Number of pairs generated to reduce the number of components from 3276800 to 1 is 25345048

Process finished with exit code 0

Unit Test

UF_HWQUPC_Test.java