Program Structures and Algorithms
Spring 2023 (SEC –3)

NAME: Kaushik Gnanasekar
NUID: 002766012

Task: Assignment 4 (WQUPC)

Relationship Conclusion:

The implementation of height-weighted Quick Union with Path Compression takes *log(n)* time to merge components. For *n* sites to be merged, it'll take *nlog(n)* time. The path compression, flattens the height from nodes to roots reducing the overall time by a factor *k*, hence we get *k * nlog(n)*.

From the tests done for the number of objects (n) and the number of pairs (m) we find the relationship is:

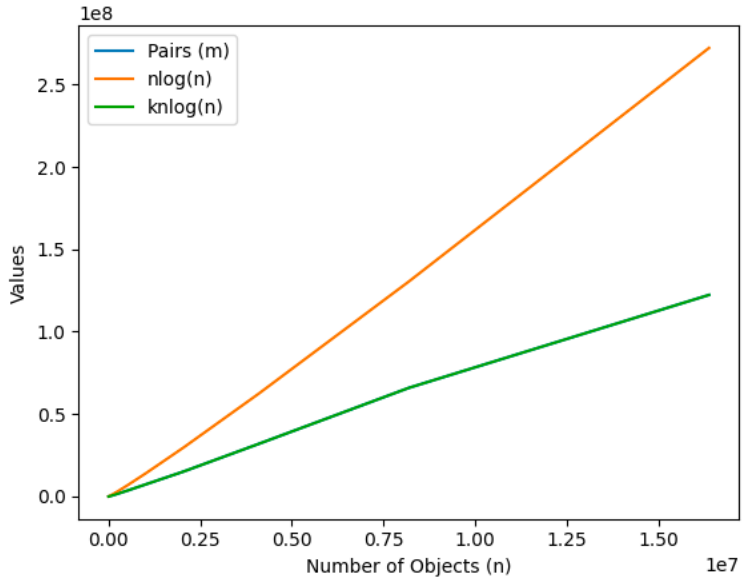
$m = nlog(n) * k$

Where k is some constant, which according to the tests done is 0.521064576.

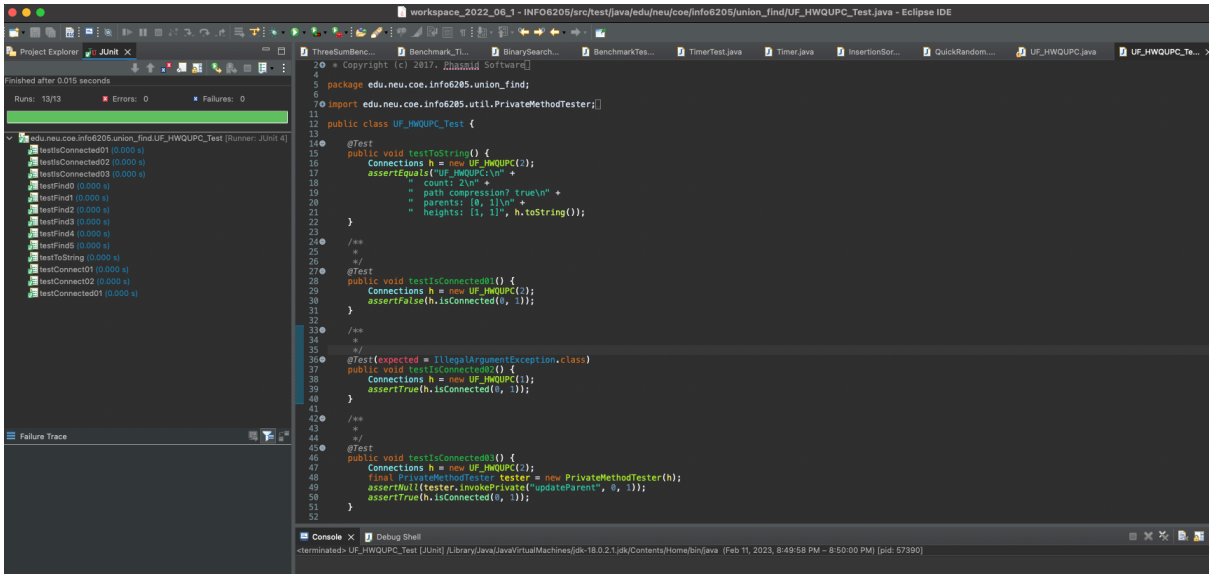
Evidence to support that conclusion:

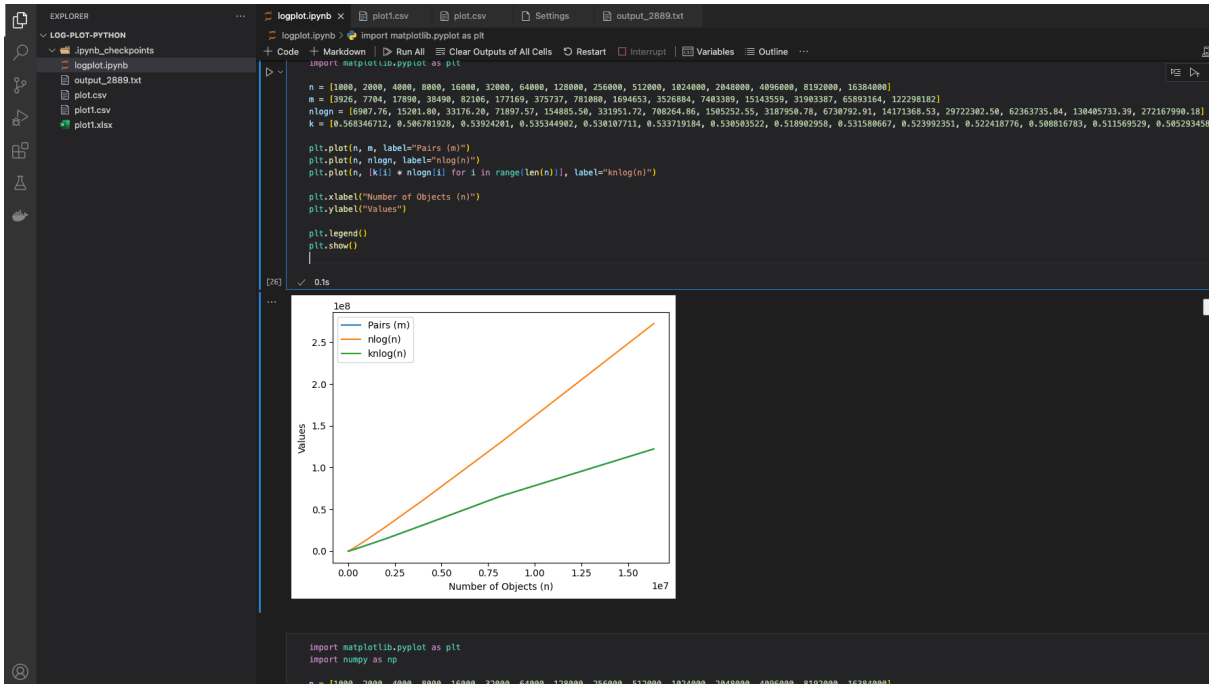
Number of Objects (n)	Number of Pairs (m)	nlog(n)	Constant (k)
1000	3926	6,907.76	0.568346712
2000	7704	15,201.80	0.506781928
4000	17890	33,176.20	0.53924201
8000	38490	71,897.57	0.535344902
16000	82106	154,885.50	0.530107711
32000	177169	331,951.72	0.533719184
64000	375737	708,264.86	0.530503522
128000	781080	1,505,252.55	0.518902958
256000	1694653	3,187,950.78	0.531580667
512000	3526884	6,730,792.91	0.523992351
1024000	7403389	14,171,368.53	0.522418776
2048000	15143559	29,762,302.50	0.508816783
4096000	31903387	62,363,735.84	0.511569529
8192000	65893164	130,405,733.39	0.505293458
16384000	122298182	272,167,990.18	0.449348147
			0.521064576

Graphical Representation:



Unit Test Screenshots:




EXPLORER
LOG-PLOT-PYTHON
.ipynb_checkpoints
logplot.ipynb
output_2889.txt
plot.csv
plot1.csv
plot1.xlsx
logplot.ipynb
plot1.csv
plot.csv
Settings
output_2889.txt
logplot.ipynb > import matplotlib.pyplot as plt
Code
Markdown
Run All
Clear Outputs of All Cells
Restart
Interrupt
Variables
Outline
import matplotlib.pyplot as plt
n = [1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000, 256000, 512000, 1024000, 2048000, 4096000, 8192000, 16384000]
m = [3926, 7704, 17890, 38490, 82106, 177169, 375737, 781080, 1694653, 3526884, 7403389, 15143559, 31903387, 65893164, 122298182]
nlogn = [6907.76, 15201.80, 33176.20, 71897.57, 154885.50, 331951.72, 708264.86, 1505252.55, 3187950.78, 6730792.91, 14171368.53, 29722302.50, 62363735.84, 130405733.39, 272167990.18]
k = [0.568346712, 0.506781928, 0.53924201, 0.535344902, 0.530107711, 0.533719184, 0.530503522, 0.518902958, 0.531580667, 0.523992351, 0.522418776, 0.508816783, 0.511569529, 0.505293458
plt.plot(n, m, label="Pairs (m)")
plt.plot(n, nlogn, label="nlog(n)")
plt.plot(n, [k[i] * nlogn[i] for i in range(len(n))], label="knlog(n)")
plt.xlabel("Number of Objects (n)")
plt.ylabel("Values")
plt.legend()
plt.show()
[76]
0.1s
1e8
Pairs (m)
nlog(n)
knlog(n)
2.5
2.0
1.5
1.0
0.5
0.0
Values
0.00
0.25
0.50
0.75
1.00
1.25
1.50
Number of Objects (n)
1e7
import matplotlib.pyplot as plt
import numpy as np