Program Structures & Algorithms

Spring 2022

Assignment Number 3

Name: Kanishk Bimalkumar Bhatia

NUID: 001580259

- Task:

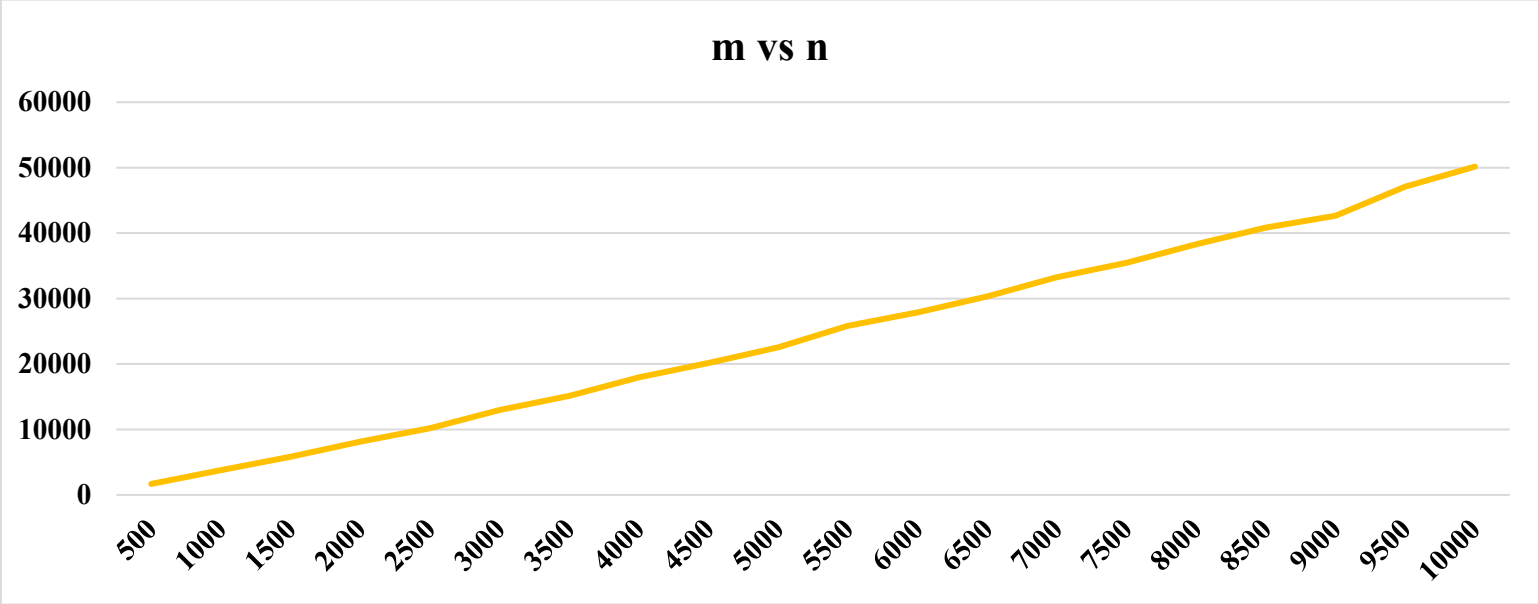
1. (a) Implement height-weighted Quick Union with Path Compression. For this, you will flesh out the class UF_HWQUPC. All you must do is to fill in the sections marked with // TO BE IMPLEMENTED ... // ...END IMPLEMENTATION.
   (b) Check that the unit tests for this class all work. You must show "green" test results in your submission (screenshot is OK).
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).
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.

- Evidence:

  The results and main functions generate the necessary data to get the relation between m and n. For every value of n, a random set of pairs are generated and checked for a connection. If they are not connected, they are sent to the union function. This process is repeated until all the pairs are connected. The number of iterations required to complete this process is “m”.

  The function runs between the range of 500 to 10000, increasing by 100 every time for a value of n, and the corresponding value of m is calculated.

Graph:



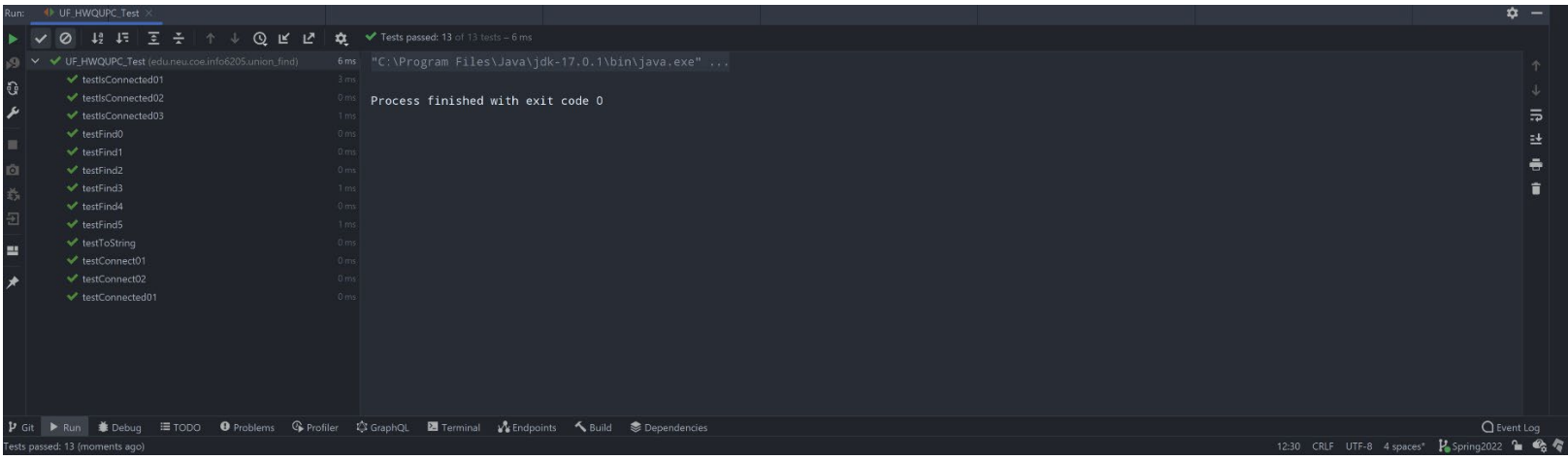
- Conclusion:

  As per the graph above, we observe that the relation between m and n is close to linear, but not exactly linear. This variation can be accounted for, by having a coefficient.

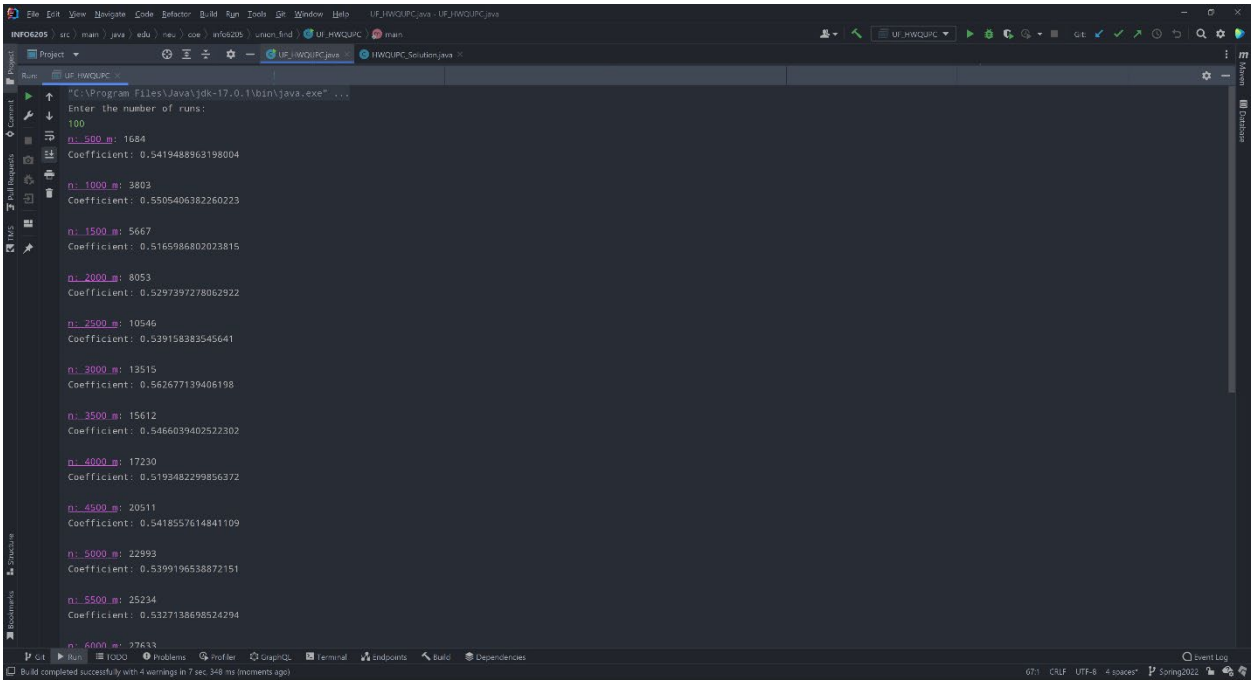
  If the maximum height of the node log(n), and the number of unions performed is n, then we deduce that **m = n * log(n)**

  The formula will be:
  m = coefficient * n * log(n)
  As per the data generated, the coefficient is approximately 0.535211823864416

- Test Screenshot:



Data:



n	m	n log n	m/n log n
500	1688	1349.485002	1.250847544
1000	3798	3000	1.266
1500	5846	4764.136889	1.227084808
2000	8127	6602.059991	1.230979423
2500	10213	8494.850022	1.202257836
3000	12974	10431.36376	1.243749168
3500	15137	12404.23816	1.220308721
4000	17958	14408.23997	1.246370136
4500	20159	16439.45631	1.226257099
5000	22544	18494.85002	1.218933918
5500	25823	20571.99479	1.255250172
6000	27910	22668.9075	1.231201812
6500	30346	24783.93682	1.224422101
7000	33263	26915.68628	1.235822102

7500	35469	29062.95948	1.220419415
8000	38286	31224.7199	1.226143905
8500	40836	33400.06087	1.222632502
9000	42631	35588.18258	1.197897642
9500	47115	37788.37425	1.24681204
10000	50161	40000	1.254025

Enter the number of runs:

100

n: 500 m: 1688

Coefficient: 0.5432361858597524

n: 1000 m: 3798

Coefficient: 0.5498168140895169

n: 1500 m: 5846

Coefficient: 0.5329161610134326

n: 2000 m: 8127

Coefficient: 0.5346075708284784

n: 2500 m: 10213

Coefficient: 0.522133943784528

n: 3000 m: 12974

Coefficient: 0.5401534004184989

n: 3500 m: 15137

Coefficient: 0.5299733438123244

n: 4000 m: 17958

Coefficient: 0.5412916723204918

n: 4500 m: 20159

Coefficient: 0.5325566913245668

n: 5000 m: 22544

Coefficient: 0.529376274398007

n: 5500 m: 25823

Coefficient: 0.5451482230799432

n: 6000 m: 27910

Coefficient: 0.5347041531970141

n: 6500 m: 30346

Coefficient: 0.5317597621605238

n: 7000 m: 33263

Coefficient: 0.5367107195861662

n: 7500 m: 35469

Coefficient: 0.5300214175244251

n: 8000 m: 38286

Coefficient: 0.5325075321592316

n: 8500 m: 40836

Coefficient: 0.5309825491871837

n: 9000 m: 42631

Coefficient: 0.5202403357862142

n: 9500 m: 47115

Coefficient: 0.5414835890892967

n: 10000 m: 50161

Coefficient: 0.5446161376687253

Average Coefficient: 0.535211823864416

Process finished with exit code 0