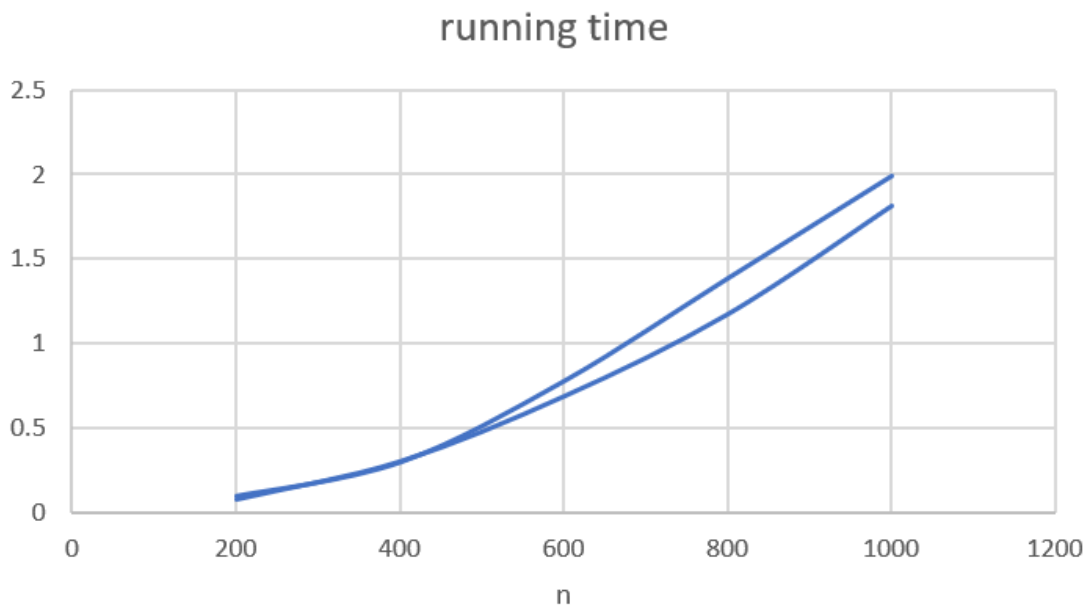
For two alternatives of Union-Find, I benchmark each of them. Using site number n in different values: 200 to 1000, I calculate the running time for weighted quick union with path compression and weighted quick union without path compression.

The result data is as follows:

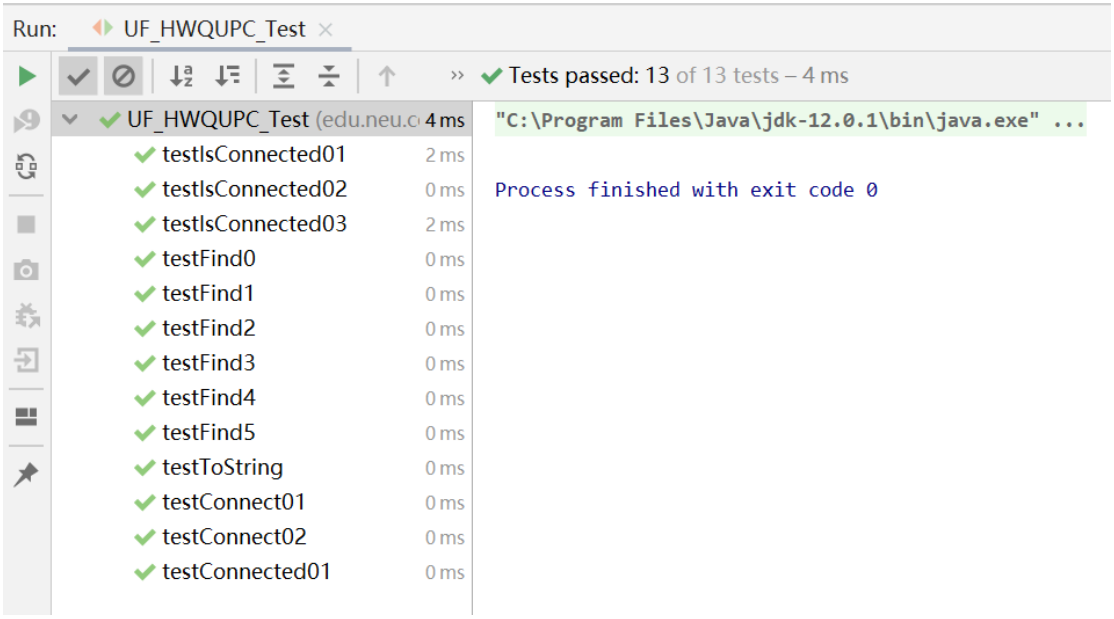
	n(sites)	running time
	1000	1.992
without path compression	800	1.387
	600	0.779
	400	0.299
	200	0.099
	1000	1.815
with path compression	800	1.175
	600	0.689
	400	0.304
	200	0.079

I also drew the line graphs for the two alternatives:



From the data and the graphs, we can see that the running time increases with the increase of n, and the running time of weighted quick union with path compression is less than weighted quick union without path compression. Since the improvement of path compression in find(), the performance of weighted quick union with path compression is obviously better than weighted quick union without path compression. And for weighted quick union with path compression, theoretically speaking, the relationship should based on the iterated log function, but according to our experimental results, it tends to be more linear, which is reasonable and convincing.

- UF_HWQUPC_Test screenshot


Run:
UF_HWQUPC_Test
Tests passed: 13 of 13 tests – 4 ms
UF_HWQUPC_Test (edu.neu.c 4 ms
testIsConnected01 2 ms
testIsConnected02 0 ms
testIsConnected03 2 ms
testFind0 0 ms
testFind1 0 ms
testFind2 0 ms
testFind3 0 ms
testFind4 0 ms
testFind5 0 ms
testToString 0 ms
testConnect01 0 ms
testConnect02 0 ms
testConnected01 0 ms
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" ...
Process finished with exit code 0