Aditya Dasani (NUID- 001560067)

**Program Structures & Algorithms**

**Spring 2021**

**Assignment No. 4**

* **Task**

For weighted quick union, store the depth rather than the size.

For weighted quick union with path compression, do two loops, so that all intermediate nodes point to the root, not just the alternates.

* **Output**

All my unit test cases of Weighted Quick Union and Union find are running. I am running the path compression till root and path compression till grandparent for different values of nodes and with doubling method, and printing the values of the time taken to make 1 tree.

* **Merging by height.**

When we merge the tree by height, the shorter tree goes and connects to the bigger one. The only time when the height will increase will be when we connect two trees of the same height. The height will be increased by 1.

The maximum height of the tree with N nodes can be at most log(n).

h<=lg(n)

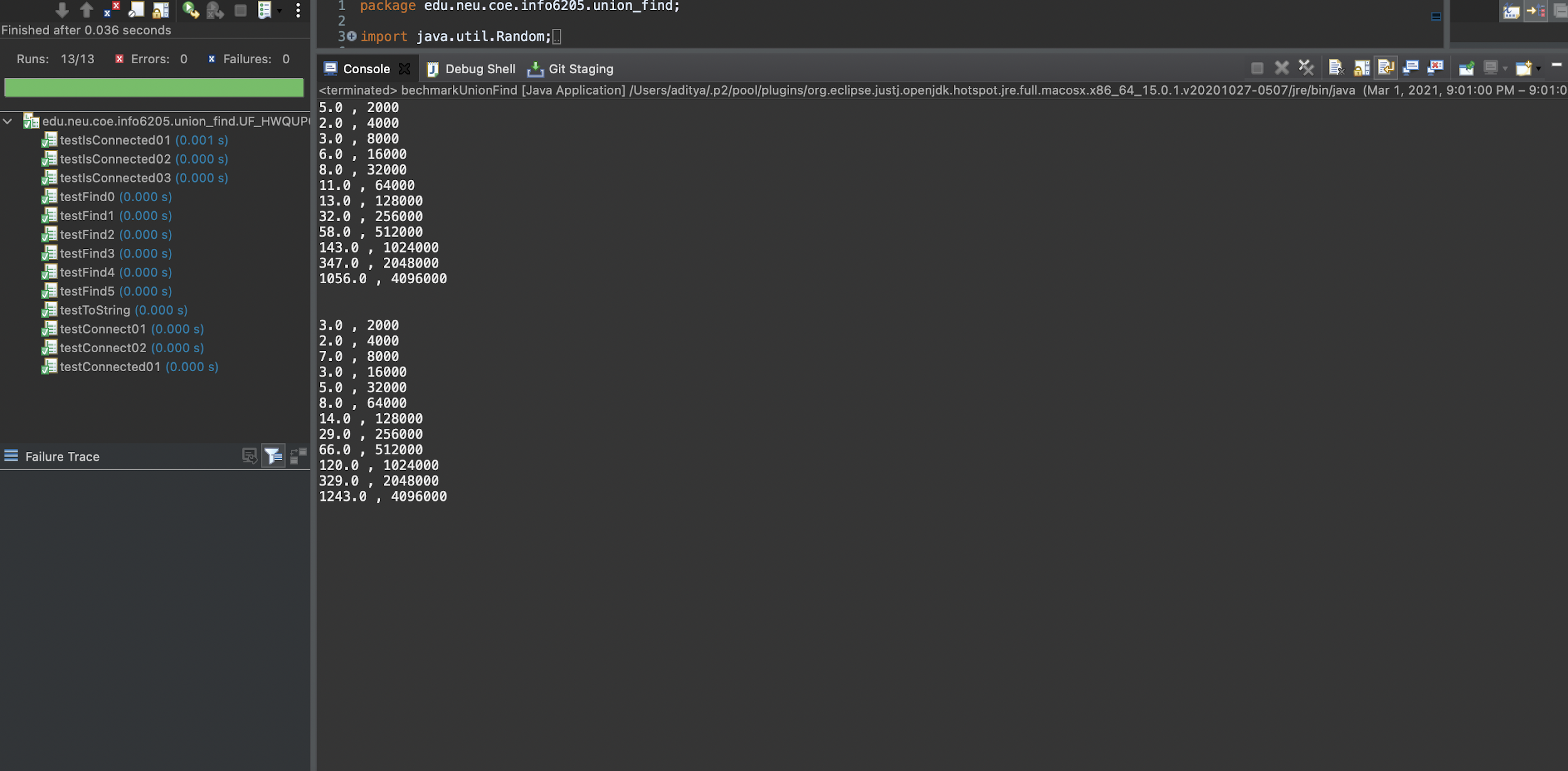
* **Merging by size.**

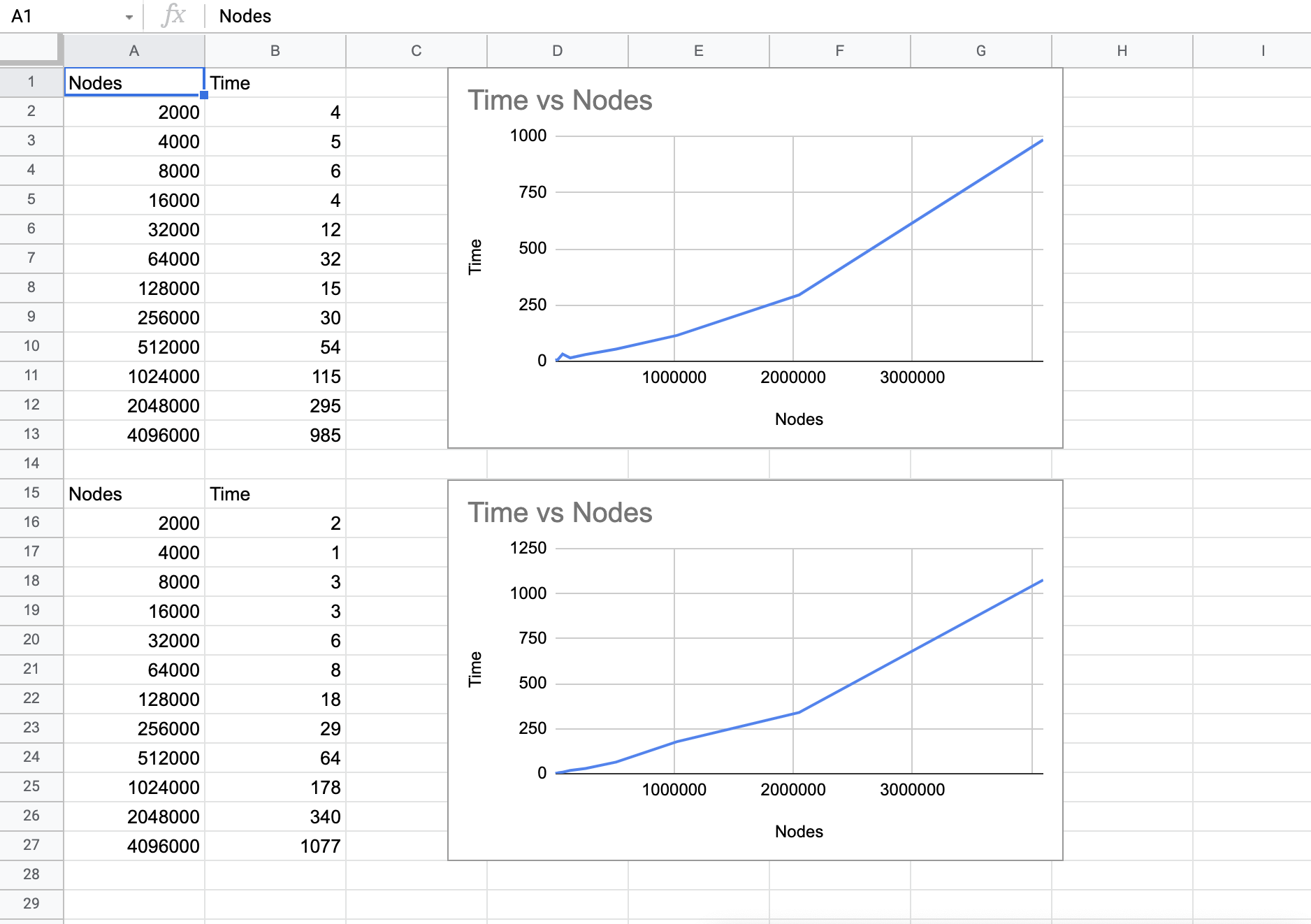
When we merge the tree by size, the tree having less nodes goes and connects to the tree having more nodes. Then we increase the size of the parent by adding the size of the child tree to it. By which we can say that height will only be increased when two trees of the same height are merged. So, when we connect all the nodes and make one parent, the size of the tree will be equal to n. Means all the nodes are in one tree having n size.

h<=lg(n)

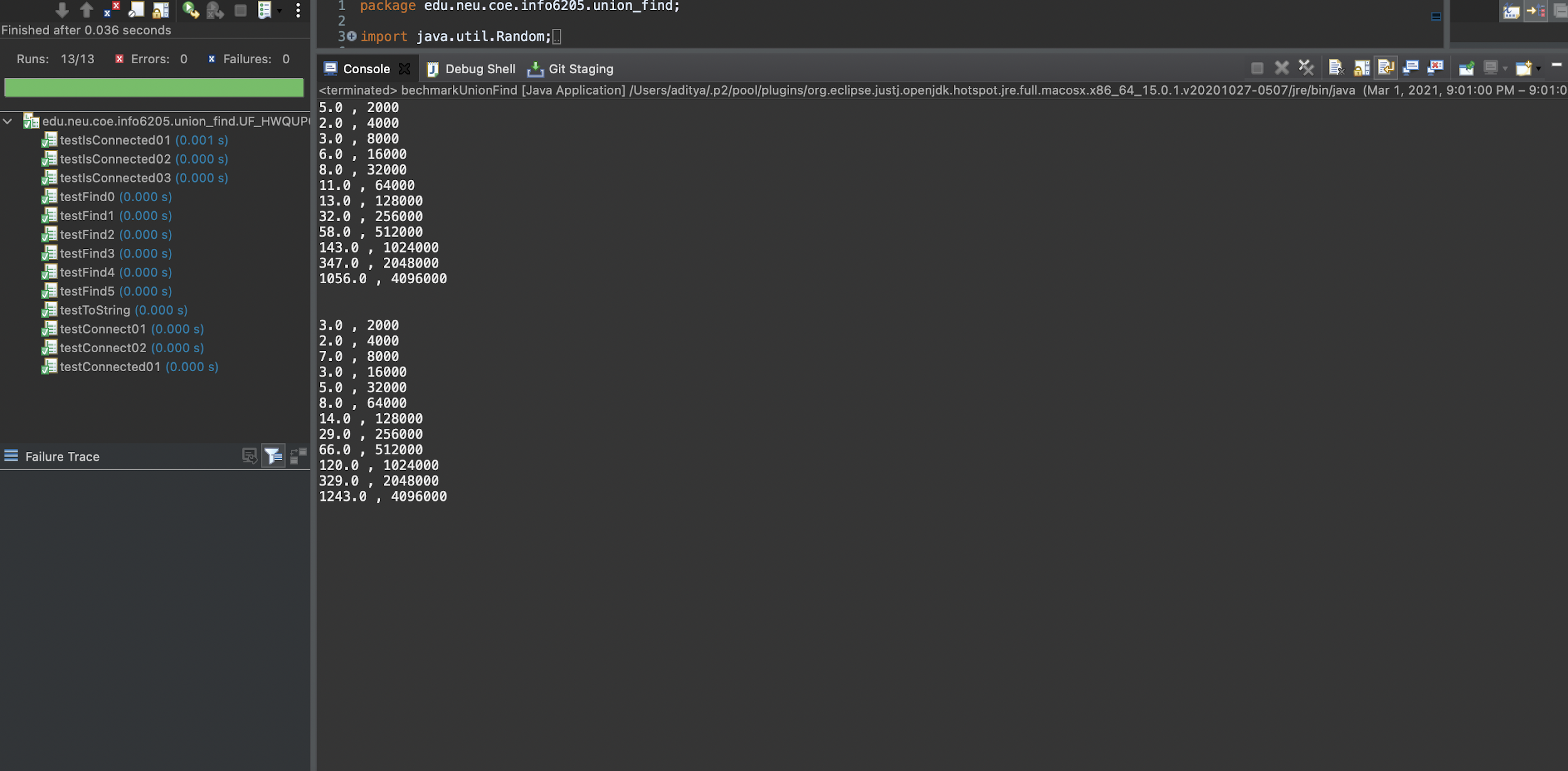
So by these relationships of size and height, they have the same upper bound. So we can say that merging by height or size will not make a difference in performance. So, benchmarking is not required.

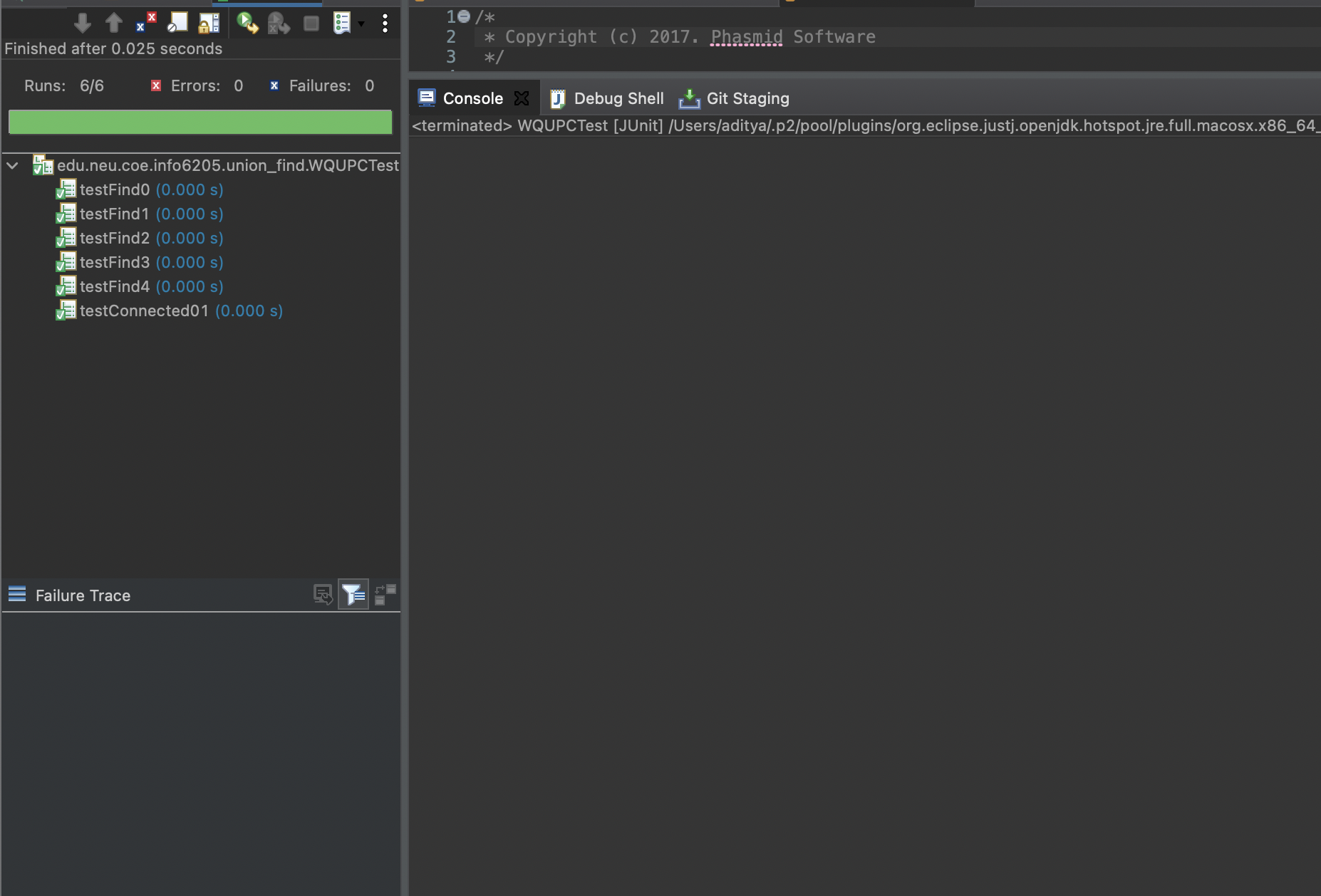
* **Evidence to support the conclusion**

****

****

* **Unit Test Result**

****

****

* **Relationship Conclusion**

The relationship between both is that time taken by full path compression is a bit less than the path compression with parent of parent.