Data Structures & Algorithms

Sep 19, 2018

Exo: 1.4.38 (Page 214)

Run experiments to evaluate the following implementation of the inner loop of ThreeSum:

Do so by developing a version of DoublingTest that computes the ratio of the running times of this program and *ThreeSum.*

Authors: Cecilia Sui, Derek Dyer, Maeve Mueller

1. Find T(2N) / T(N) = 2^b for the Naive 3-sum implementation. In this case, the naive 3-sum implementation has a time complexity of O(n^3). Thus, b = 3, which means the value from the data generated should converge to 3. When we repeated our Doubling Test for Naive 3-sum implementation, our T(2N)/ T(N) did converge to a number that gives us a convergence to 3 for the b value. Thus, the data proves that the complexity is of O(n^3).

Please see the source code in DoublingTest.java

N	Time1 (ns)	T(2N) / T(N) = 2^b	Estimated b	Time2	2: 2^b	Estimated b2
1	0.000000	0.000000	0.000	0.000000	0.000000	0.000
2	0.000000	0.000000	0.000	0.000000	0.000000	0.000
4	0.000000	0.000000	0.000	0.000000	0.000000	0.000
8	0.000000	0.000000	0.000	0.000000	0.000000	0.000
16	0.000000	0.000000	0.000	0.000000	0.000000	0.000

32	0.000001	0.000000	0.000	0.000001	0.000000	0.000
64	0.000004	4.000000	2.000	0.000005	5.000000	2.322
128	0.000007	1.750000	0.807	0.000008	1.600000	0.678
256	0.00004	5.714286	2.515	0.000037	4.625000	2.209
512	0.000062	1.550000	0.632	0.000086	2.324324	1.217
1024	0.000468	7.548387	2.916	0.000504	5.860465	2.551
2048	0.003811	8.143162	3.026	0.005633	11.176587	3.482
4096	0.034292	8.998163	3.170	0.035740	6.344754	2.666
8192	0.276869	8.073866	3.013	0.269436	7.538780	2.914

2. Please see the chart below for the time elapsed for Naive-Three-Sum implementation and the Three-Sum implementation given in the textbook, along with the ratio between the two (Naive-Three-Sum / Three-Sum)

Please see the source code in CompareThreeSum.java

The ratio converges to 4.

N	NaiveThreeSum	ThreeSum	Ratio(NTS/TS)
1	0.000000	0.000001	0.000000
2	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000

32	0.000001	0.000001	1.000000
64	0.000003	0.000002	1.500000
128	0.000006	0.000004	1.500000
256	0.000051	0.000012	4.250000
512	0.000074	0.000019	3.894737
1024	0.000600	0.000126	4.761905
2048	0.003944	0.000958	4.116910
4096	0.038108	0.009810	3.884608
8192	0.311992	0.077363	4.032832