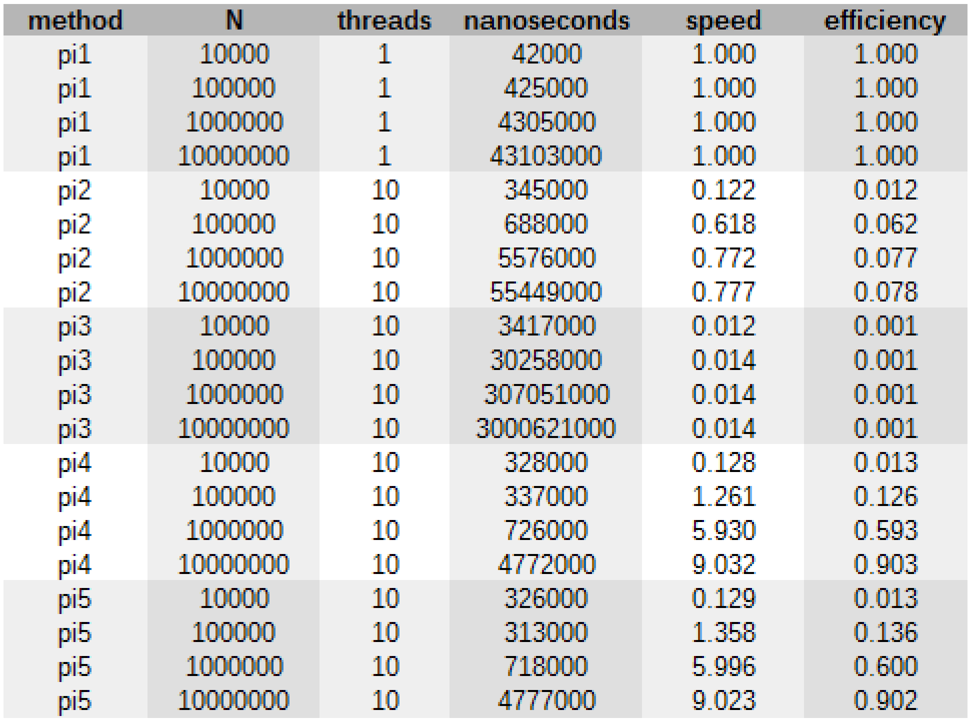
Program 1: pthread Pi Approximation

Thomas Hughes

CSC-4760, Fall 2015



method	N	threads	nanoseconds	speed	efficiency
pi1	10000	1	42000	1.000	1.000
pi1	100000	1	425000	1.000	1.000
pi1	1000000	1	4305000	1.000	1.000
pi1	10000000	1	43103000	1.000	1.000
pi2	10000	10	345000	0.122	0.012
pi2	100000	10	688000	0.618	0.062
pi2	1000000	10	5576000	0.772	0.077
pi2	10000000	10	55449000	0.777	0.078
pi3	10000	10	3417000	0.012	0.001
pi3	100000	10	30258000	0.014	0.001
pi3	1000000	10	307051000	0.014	0.001
pi3	10000000	10	3000621000	0.014	0.001
pi4	10000	10	328000	0.128	0.013
pi4	100000	10	337000	1.261	0.126
pi4	1000000	10	726000	5.930	0.593
pi4	10000000	10	4772000	9.032	0.903
pi5	10000	10	326000	0.129	0.013
pi5	100000	10	313000	1.358	0.136
pi5	1000000	10	718000	5.996	0.600
pi5	10000000	10	4777000	9.023	0.902

- pi1: serial
- pi2: global sum, no locking
- pi3: global sum, busy-waiting
- pi4: private sum, busy-waiting
- pi5: private sum, mutex

The results indicate that busy-wait private sum and mutex private sum methods are the fastest methods for N > 10000. I found the assignment fairly easy to complete, the only slip-up I had was that not turning off an optimization flag messed up busy-waiting at first. Almost all of the implementation is in the book, aside from timing which I used chrono for.