Kamil Gumienny Intro to Algorithms – CS2223

Project 1 - Finding the Greatest Common Divisor (GCD)

Executive Summary

Three methods were used to compute the GCD; Euler's Method, Consecutive Integer Checking (CIC), and Middle School Method (MSM). The project was coded and executed using the Python 3.6 programming language and on a machine with a 2.6 GHz i7 processor and memory running at 2133 MHz. I learned that even though it appeared as though MSM would take the longest because of a double for-loop, CIC is in fact the least effective method despite being a single while loop. After testing all three methods, it becomes apparent Euler's method is the most efficient because it is simple computations involving no loops. CIC is faster than middle school method at very small values, but it is the least efficient method for larger numbers because it has to do a computation decrementing by 1 from the smaller of the two integers until a GCD is found. CIC is the brute force way of finding GCD. Lastly, MSM is a middle of the road algorithm that computes GCD at some multiple of the time as Euler's method.

	Euler's Method	CIC	MSM
Time Efficiency	O(1)	O(n)	O(n)
Space Efficiency	O(1)	O(1)	O(log n)
Comments:	Simple Computations. Number of variables doesn't change	Poor efficiency but does not change number of variables.	Creates a new list. Worst case is dividing number by 2 till 1.

			All Times Reported in **Microseconds**				
Value "M"	Value "N"	Euler's Time	CIC Time	MSM Time	GCD	Slowest-Fastest	Comments
31415	14142	4.22	186.31	29.04	3	182.09	
4123	123	3.14	17.12	12.66	1	13.98	
15	3	2.72	6.71	8.42	3	5.7	Simple Problem
2453	5324	4.17	310.9	15.02	11	306.73	
2312	312421	3.94	186.57	25.53	1	182.63	
1232	412423	3.61	155.3	49.75	11	151.69	
4234532	2342	3.23	305.19	200.44	2	301.96	
4235132	7657576	5.09	399027.94	178.23	4	399022.85	
74567436	75685684	4.27	7292917.3	499.28	4	7292913.03	Over 7 second for GIC
3456347	3456347	2.62	4.23	160.94	3456347	158.32	Best Case Scenario