SUMMARY

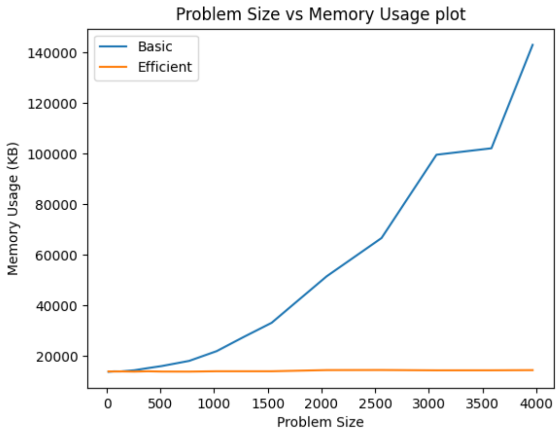
USC ID/s: 7870573217, 6565038173, 6117696230

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.0	0.0	13864.0	14000.0
64	1.5869140625	3.0126571655273438	14032.0	13968.0
128	12.833833694458008	11.99960708618164	14008.0	14004.0
256	136.80315017700195	62.22653388977051	14468.0	13936.0
384	50.5366325378418	100.6472110748291	15328.0	14060.0
512	139.55116271972656	199.99265670776367	16132.0	13952.0
768	187.43276596069336	447.800874710083	18176.0	13932.0
1024	362.05196380615234	682.3165416717529	22020.0	14076.0
1280	577.0320892333984	1037.4088287353516	27716.0	14072.0
1536	849.576473236084	1520.9853649139404	33220.0	14072.0
2048	1732.1856021881104	2751.5475749969482	51564.0	14532.0
2560	2372.6236820220947	4557.240724563599	66712.0	14568.0
3072	3737.927198410034	7852.494955062866	99628.0	14440.0
3584	3846.257448196411	13125.321865081787	102176.0	14452.0
3968	6143.084287643433	18215.56043624878	143044.0	14520.0

Insights

Graph1 – Memory vs Problem Size (M+N)



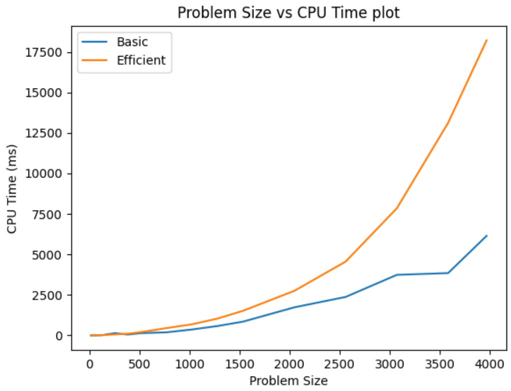
Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)

Basic: Polynomial

Efficient: Linear

Explanation: When input size increases the memory usage difference between the two programs is most noticeable. For the memory efficient algorithm, values can be computed from only the previously computed row, meaning its memory requirements are way less. The basic algorithm grows at a rate of m*n.

Graph2 – Time vs Problem Size (M+N)



Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)

Basic: Polynomial

Efficient: Polynomial

Explanation: Although both the basic and efficient algorithms run in O(m*n) time the efficient algorithm shows higher times, which are especially noticeable in larger problem sizes. This is because during divide and conquer, the efficient version performs computations at every level.

Contribution

7870573217: Equal Contribution

6565038173: Equal Contribution

6117696230: Equal Contribution