SUMMARY

USC ID/s: 7549953295, 8029126041

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.09202957153320312	0.14901161193847656	12144	12704
64	0.5078315734863281	1.0411739349365234	12112	12240
128	1.6410350799560547	3.5576820373535156	12608	12592
256	6.455898284912109	13.203144073486328	12320	12176
384	11.8560791015625	27.04787254333496	12592	12448
512	23.590803146362305	47.68109321594238	13056	12352
768	51.917076110839844	102.76198387145996	14080	12640
1024	90.71826934814453	175.5967140197754	14752	12512
1280	141.0822868347168	277.63986587524414	16416	12448
1536	200.85883140563965	391.6471004486084	18208	12816
2048	374.1738796234131	712.7070426940918	22224	12576
2560	582.042932510376	1100.7189750671387	26864	12608
3072	837.7461433410645	1587.601900100708	23792	12720
3584	1142.5018310546875	2189.5229816436768	41520	12288
3968	1418.8122749328613	2690.8621788024902	34480	12992

Insights

Graph1 – Memory vs Problem Size (M+N)
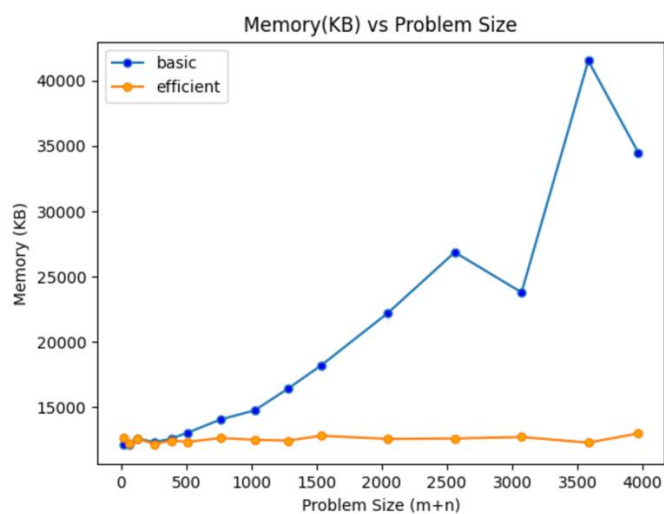
Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)
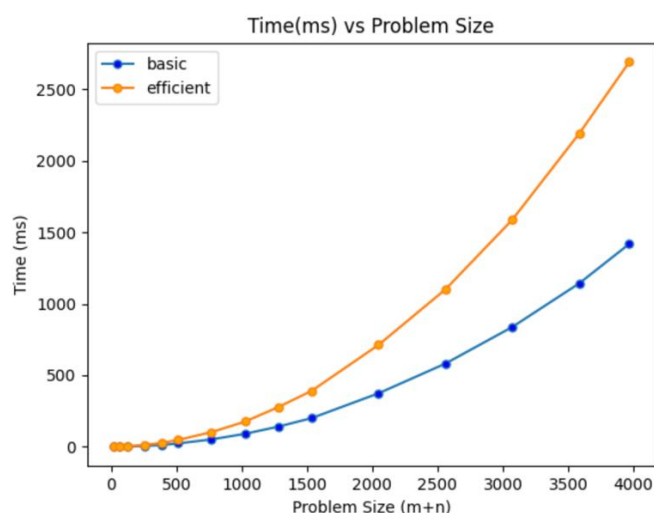Basic: Exponential
Efficient: Constant

The blue line is the graph for the basic method, and the orange line is the graph for the efficient method.

The memory for the basic method increases exponentially as problem size increases, indicating that it requires significantly more memory to handle larger inputs. This is because the traditional dynamic programming approaches in sequence alignment have a space complexity that is positively affected by the sum of the lengths of the two sequences being aligned.

On the other hand, the memory for the efficient method almost stays constant no matter what the problem size is. This is because using a divide-and-conquer technique limits memory consumption by solving smaller sub-problems independently.

Graph2 – Time vs Problem Size (M+N)

Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)
Basic: Polynomial
Efficient: Polynomial

Explanation:
The blue line is the graph for the basic method, and the orange line is the graph for the efficient method.

Both methods increase in polynomial time. However, as the problem size increases, the time used by the efficient method grows faster than that by the basic method. This is because the efficient method splits the problem into several sub-problems to solve, which requires increasing time as the problem size increases.

Contribution
8029126041: Equal Contribution
7549953295: Equal Contribution