SUMMARY

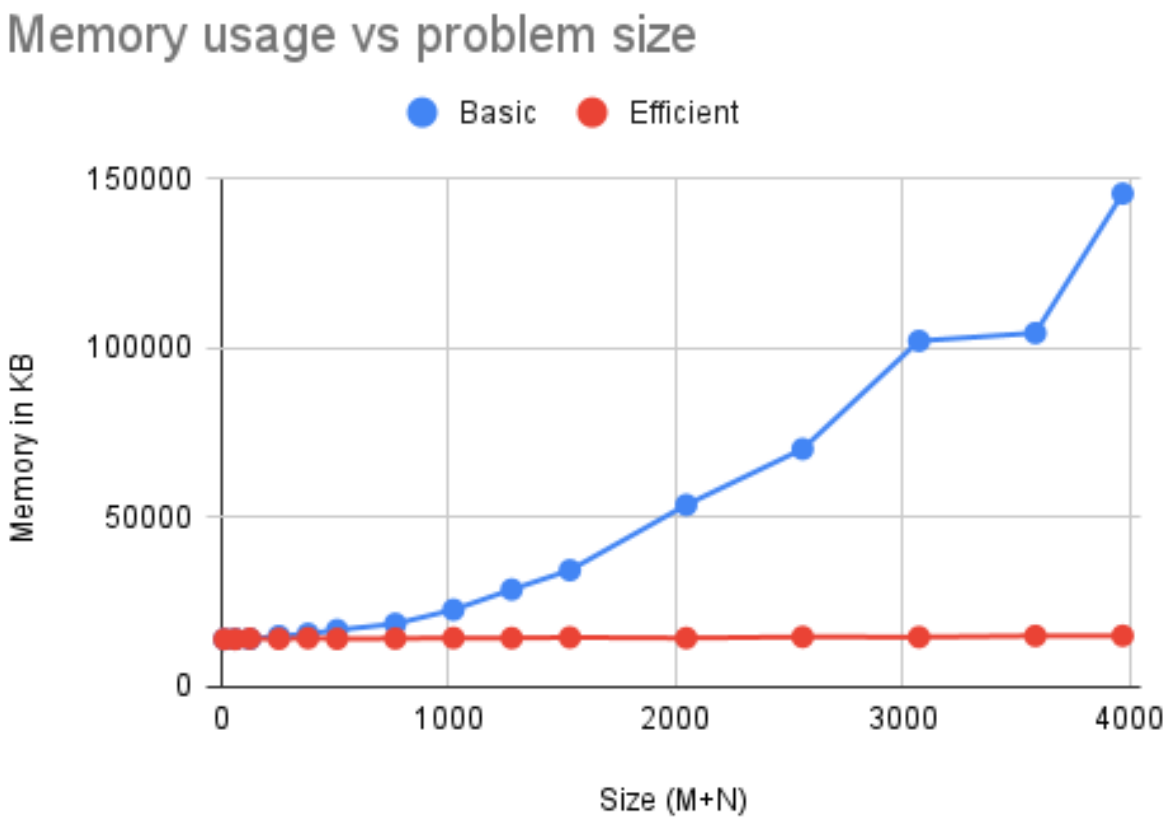
USC ID/s: 4916916532, 8848045890

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.00	0.00	13908	13916
64	1.99	5.178	13976	13916
128	4.99	12.041	14000	14092
256	21.94	52.328	14844	14004
384	46.90	105.613	15512	14220
512	80.78	180.819	16620	14024
768	175.53	405.024	18460	14120
1024	353.10	699.500	22608	14284
1280	546.54	1126.452	28572	14248
1536	905.60	1764.673	34276	14440
2048	1505.99	2738.683	53556	14252
2560	2123.47	4718.462	70060	14600
3072	3308.19	6719.891	101968	14536
3584	4193.76	8905.214	104292	14896
3968	5243.01	11051.276	145456	14936

Insights

Graph1 – Memory vs Problem Size (M+N)



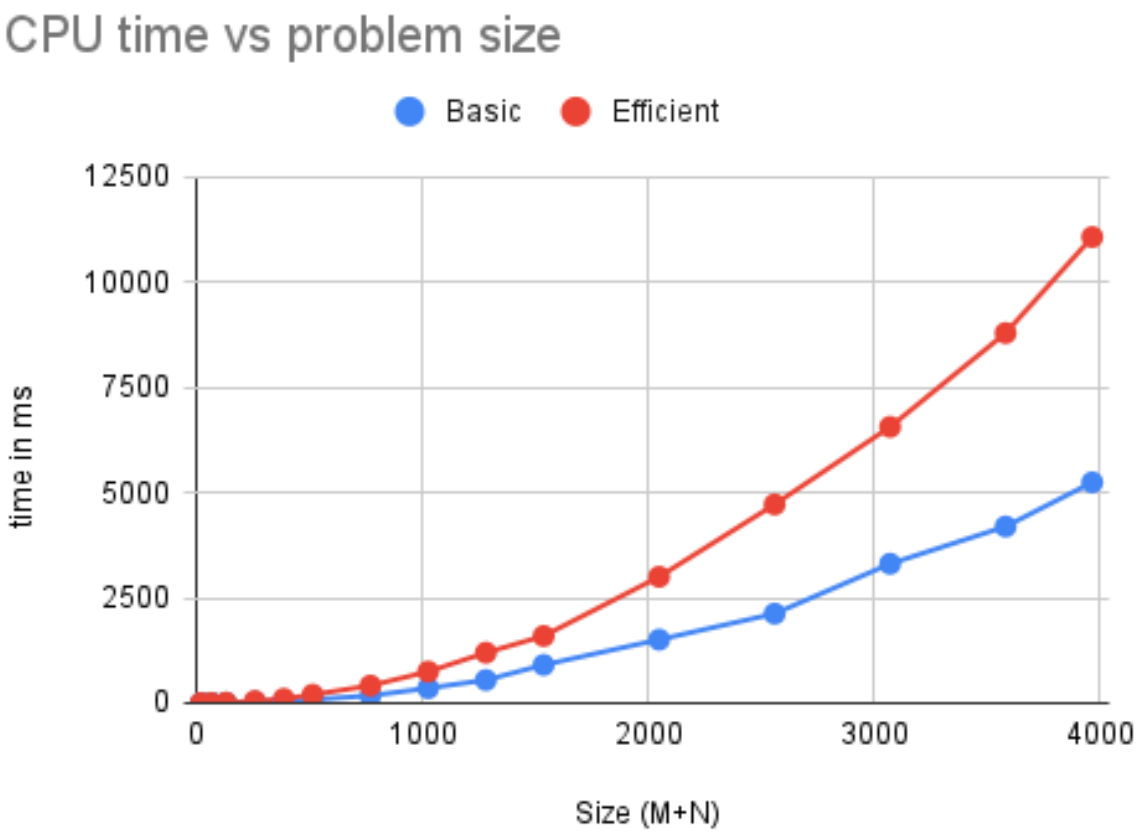
Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)

Basic: Polynomial
Efficient: Linear

Explanation:

The Memory vs Size graph represents the memory consumption of the efficient and basic algorithm across various problem sizes. The efficient algorithm plot is linear and close to the x axis whereas, the basic algorithm's plot is polynomially increasing. The basic algorithm needs O(m*n) memory and the efficient algorithm needs O(m+n) memory. In the space efficient solution, we just need 2 columns in the subproblems, Hence space complexity is linear, whereas the basic solution requires a 2 dimensional array therefore it needs polynomial memory.

Graph2 – Time vs Problem Size (M+N)



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

Basic: Polynomial

Efficient: Polynomial

Explanation:

The Time vs Size graph represents the Time taken by the efficient and basic algorithm across various problem sizes. The efficient algorithm plot is polynomially increasing and the basic algorithm's plot is also polynomially increasing. The basic algorithm needs O(m*n) time and the efficient algorithm also needs O(m*n) time. The efficient algorithm's line is more steeper as it takes more time with increasing input size compared to the basic algorithm. The efficient algorithm takes roughly twice the time of the basic algorithm.

Contribution

4916916532 : Equal Contribution

8848045890: Equal Contribution