**SUMMARY**

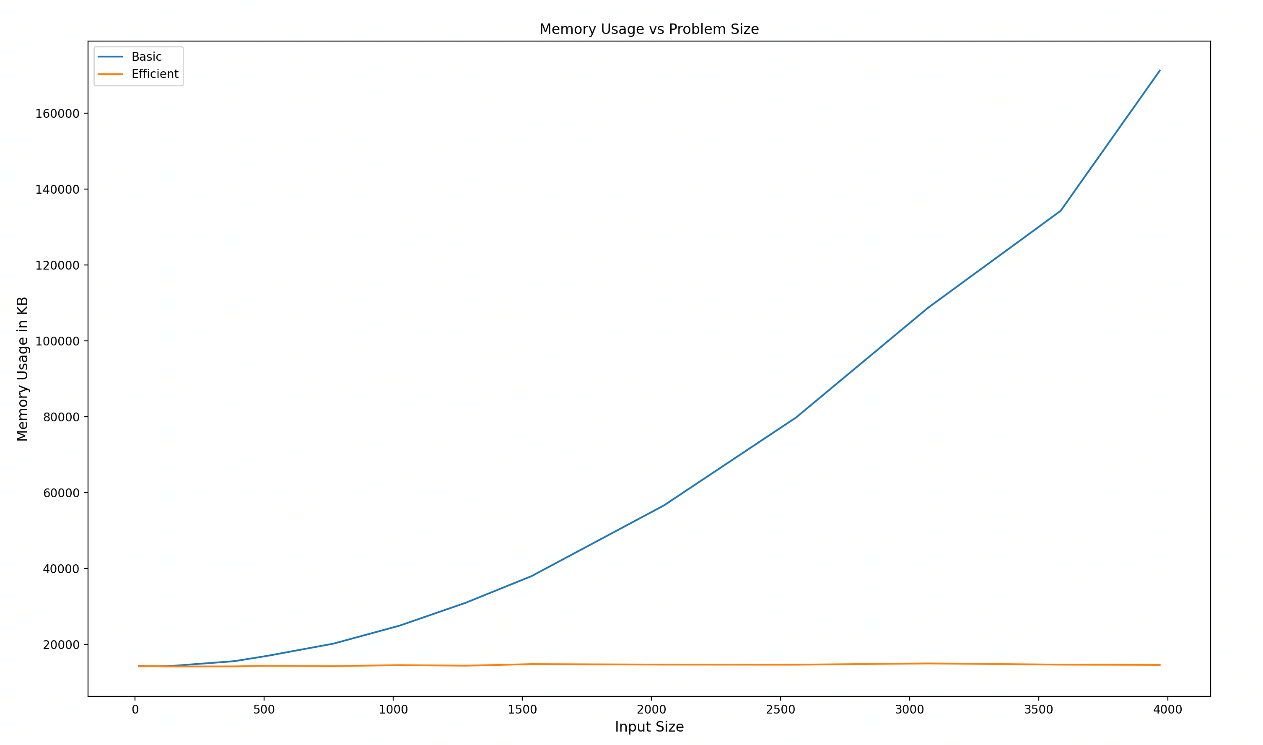
## USC ID/s: 5061063668, 5536870165, 6531106261

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M+N | Time in MS (Basic) | Time in MS (Efficient) | Memory in KB (Basic) | Memory in KB (Efficient) |
| 16 | 0.3409385681152344 | 0.2918243408203125 | 14664 | 14620 |
| 64 | 1.7368793487548828 | 1.7290115356445312 | 14828 | 14748 |
| 128 | 3.5381317138671875 | 5.688190460205078 | 14592 | 14980 |
| 256 | 10.319232940673828 | 21.143198013305664 | 15732 | 15052 |
| 384 | 23.074865341186523 | 44.71182823181152 | 16372 | 14960 |
| 512 | 39.34907913208008 | 80.1088809967041 | 17568 | 15020 |
| 768 | 90.43097496032715 | 176.79905891418457 | 20784 | 15132 |
| 1024 | 163.04826736450195 | 315.81997871398926 | 25520 | 14916 |
| 1280 | 260.1339817047119 | 506.0086250305176 | 31432 | 15040 |
| 1536 | 380.65481185913086 | 712.4888896942139 | 38360 | 15080 |
| 2048 | 677.9561042785645 | 1293.9348220825195 | 56928 | 15320 |
| 2560 | 1065.8738613128662 | 2094.317674636841 | 80592 | 15384 |
| 3072 | 1521.200180053711 | 2805.553913116455 | 109420 | 15412 |
| 3584 | 2061.429023742676 | 3717.4320220947266 | 126848 | 15540 |
| 3968 | 2553.7259578704834 | 4836.273193359375 | 155484 | 15200 |

## Datapoints

## Insights

### Graph1 – Memory vs Problem Size (M+N)



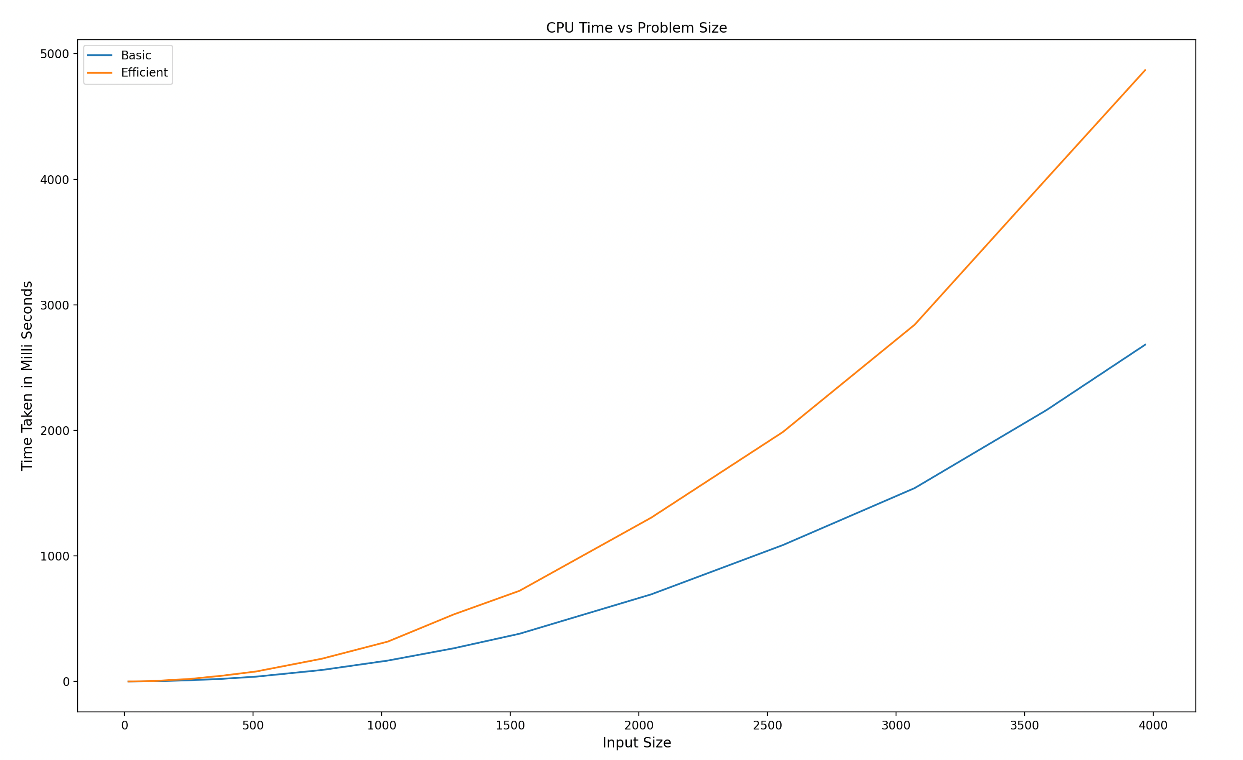
#### Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)

Basic: Polynomial

Efficient: Linear

#### Explanation: Basic Algorithm uses O(m\*n) space whereas the Efficient Algorithm uses O(min(m, n)) space since we are just storing 2 columns of data and reusing those columns as we move forward with our algorithm. Thus, the efficient algorithm is linear in terms of the input size whereas the basic algorithm is polynomial in terms of the input size.

### Graph2 – Time vs Problem Size (M+N)



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

Basic: Polynomial

Efficient: Polynomial

#### Explanation: Both the basic and efficient algorithms take polynomial time in terms of the input size. However, the number of computations required by the efficient version is much more as compared to the basic version since the sequence alignment needs to be figured out via the divide and conquer algorithm.

## Contribution

5061063668: Equal Contribution

5536870165: Equal Contribution

6531106261: Equal Contribution