**SUMMARY**

## USC ID/s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M+N | Time in MS (Basic) | Time in MS (Efficient) | Memory in KB (Basic) | Memory in KB (Efficient) |
| 16 | 0.1408 | 0.14381 | 153.384 | 153.384 |
| 64 | 0.9629 | 3.10554 | 153.384 | 153.384 |
| 128 | 1.93146 | 3.13595 | 306.744 | 460.104 |
| 256 | 3.88557 | 14.8277 | 1128.1 | 1840.7 |
| 384 | 26.7662 | 29.6349 | 2252.56 | 3939.78 |
| 512 | 18.1783 | 50.863 | 3999.48 | 6753.9 |
| 768 | 8.29524 | 34.8136 | 8752.62 | 6674.89 |
| 1024 | 33.1677 | 26.7539 | 2860.06 | 5394.26 |
| 1280 | 7.70463 | 14.0555 | 16359.8 | 7984.07 |
| 1536 | 69.1464 | 38.7159 | 22434.3 | 14676.3 |
| 2048 | 96.2338 | 122.735 | 16853.9 | 44957.4 |
| 2560 | 105.39 | 145.163 | 14689.2 | 17859.7 |
| 3072 | 99.8069 | 140.988 | 24882.5 | 82862.5 |
| 3584 | 190.693 | 195.929 | 64657.3 | 66119.2 |
| 3968 | 150.312 | 269.605 | 44189.5 | 67199 |

## Datapoints

## Insights

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

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

Basic: Polynomial

Efficient: Polynomial

#### Explanation:

Based on the graph above, the Basic and Efficient approach have similar memory utilization in low sequence sizes. As the size of the sequences increase, we observe a polynomial increase in both methods. However, it is observed that the Efficient approach is utilizing more memory than the Basic approach at size 2000+. Theoretically, the Basic approach and the Efficient Approach both have a run time and space complexity of O(mn) where m and n is the size of each sequence. However, the Efficient approach only uses O(m+n) space to solve the Sequence Alignment Problem. To test the validity of the theory, an increase in data points was tested (see graph below). The graph below supports the theory that increasing the input sequence size will diverge the two methods. The Basic approach will continue to use a space complexity of O(mn), while the Efficient space complexity is O(m+n) therefore showing a more linear pattern.

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

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

Basic: Polynomial

Efficient: Polynomial

#### Explanation:

A similar polynomial pattern was observed for time complexity for both approaches. Similar to memory usage, the time grows with an increase of input size due to memory allocation. Allocating memory requires time to copy memory from cache, to disk, and back again which significantly slows the programs. With an increased input size, the reading and writing operations can significantly impact the program.

## Contribution

(Please mention what each member did if you think everyone in the group does not have an equal contribution, otherwise, write “Equal Contribution”)

<USC ID/s>: <Equal Contribution>