# SUMMARY

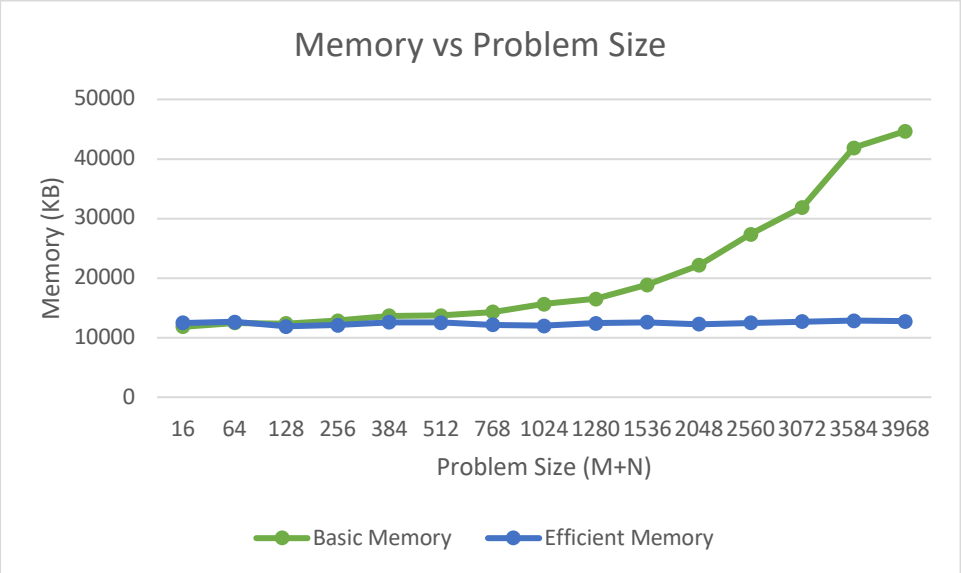
USC ID: 4858298719

## Datapoints

| M+N | Time in MS (Basic) | Time in MS (Efficient) | Memory in KB (Basic) | Memory in KB (Efficient) |
|---|---|---|---|---|
| 16 | 0.051975250244140625 | 0.1049041748046875 | 11816 | 12496 |
| 64 | 0.47206878662109375 | 0.8776187896728516 | 12496 | 12636 |
| 128 | 1.7960071563720703 | 3.164052963256836 | 12360 | 11904 |
| 256 | 6.686925888061523 | 11.3067626953125 | 12864 | 12076 |
| 384 | 15.285968780517578 | 25.389909744262695 | 13648 | 12580 |
| 512 | 24.68395233154297 | 40.7559871673584 | 13732 | 12520 |
| 768 | 55.94134330749512 | 91.64190292358398 | 14312 | 12160 |
| 1024 | 98.18387031555176 | 166.34106636047363 | 15624 | 11992 |
| 1280 | 175.41813850402832 | 263.87596130371094 | 16492 | 12424 |
| 1536 | 266.6192054748535 | 394.8700428009033 | 18868 | 12588 |
| 2048 | 427.5631904602051 | 698.7409591674805 | 22152 | 12280 |
| 2560 | 683.323860168457 | 1067.8069591522217 | 27392 | 12476 |
| 3072 | 983.3829402923584 | 1486.3018989562988 | 31880 | 12692 |
| 3584 | 1346.4970588684082 | 1992.3992156982422 | 41896 | 12868 |
| 3968 | 1712.5568389892578 | 2527.4500846862793 | 44672 | 12736 |

## Insights

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



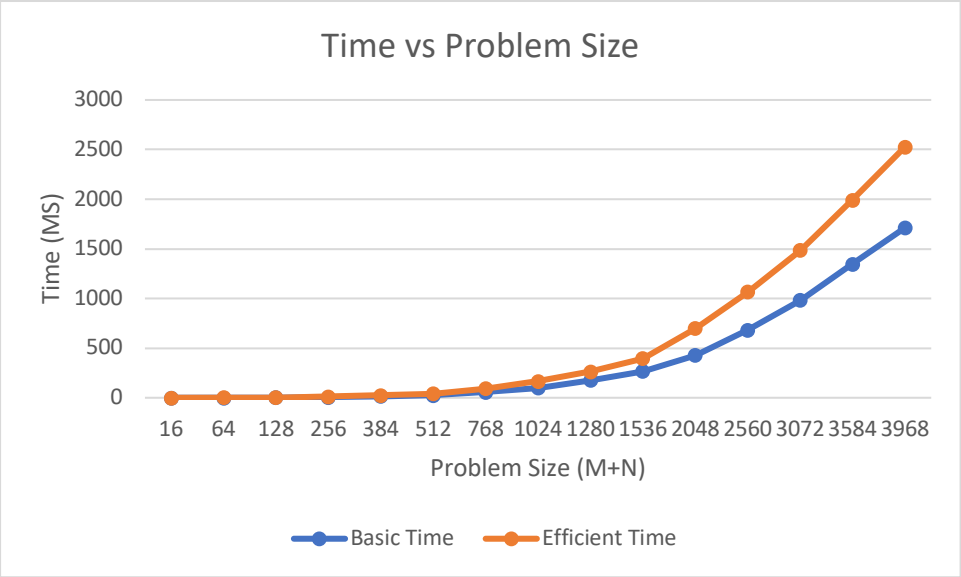
*Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)*

Basic: Polynomial

Efficient: Linear

*Explanation: The basic algorithm uses a dynamic programming solution to calculate the cost of alignment at ever combination of inputs s and t. Therefore the dp array needed to fill in all of those costs is M*N in size or polynomial with respect to the inputs M and N. On the other hand, the efficient algorithm uses a trick to only allocate 2 arrays of size N and then "fill out" the same dp costs by only using the two arrays column by column instead of the entire M*N array. Therefore the efficient is linear in respect to the input N in terms of memory.*

Graph2 – Time vs Problem Size (M+N)




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

Basic: Polynomial

Efficient: Polynomial

*Explanation: Both algorithms are polynomial as they are related to the input sizes M and N with both having a O(M*N) time complexity. However the difference in the plotted lines occurs because if the basic algorithm had a C*M*N time complexity, the memory efficient algorithm would then have a 2*C*M*N time complexity. The divide step in efficient leads to double the needed time to find the solution as each divide step level requires solving a subproblem of ½ size of the previous level. Example: C*M*N, 0.5*C*M*N, 0.25*C*M*N, … which when summed = 2*C*M*N.*

## Contribution

One man team.