SUMMARY

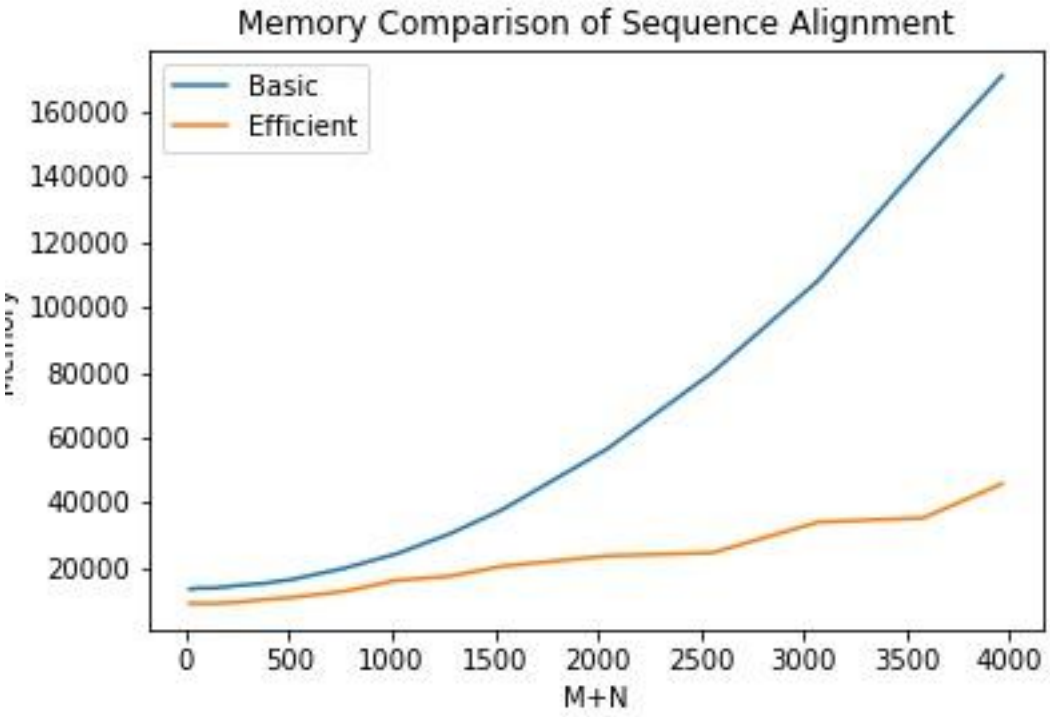
USC ID/s: 8799851341, 3598536223, 3972192394

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.2875328063964844	0.9319782257080078	13556	9136
64	0.9279251098632812	3.0777454376220703	13848	9128
128	2.3429393768310547	11.13581657409668	13816	9120
256	11.66987419128418	26.996135711669922	14668	9512
384	21.420001983642578	56.02526664733887	15336	10380
512	36.009788513183594	109.24291610717773	16468	10940
768	82.72790908813477	229.36296463012695	19924	12844
1024	144.90604400634766	394.6070671081543	24400	16228
1280	229.77614402770996	617.2301769256592	30460	17468
1536	333.32300186157227	885.5328559875488	37908	20536
2048	596.0359573364258	1609.7149848937988	56596	23740
2560	922.8541851043701	2577.008008956909	80088	24688
3072	1288.243055343628	3664.4351482391357	108160	34096
3584	1888.861894607544	5143.776893615723	144760	35380
3968	2282.579183578491	6257.5061321258545	171128	45840

Insights

Graph1 – Memory vs Problem Size (M+N)



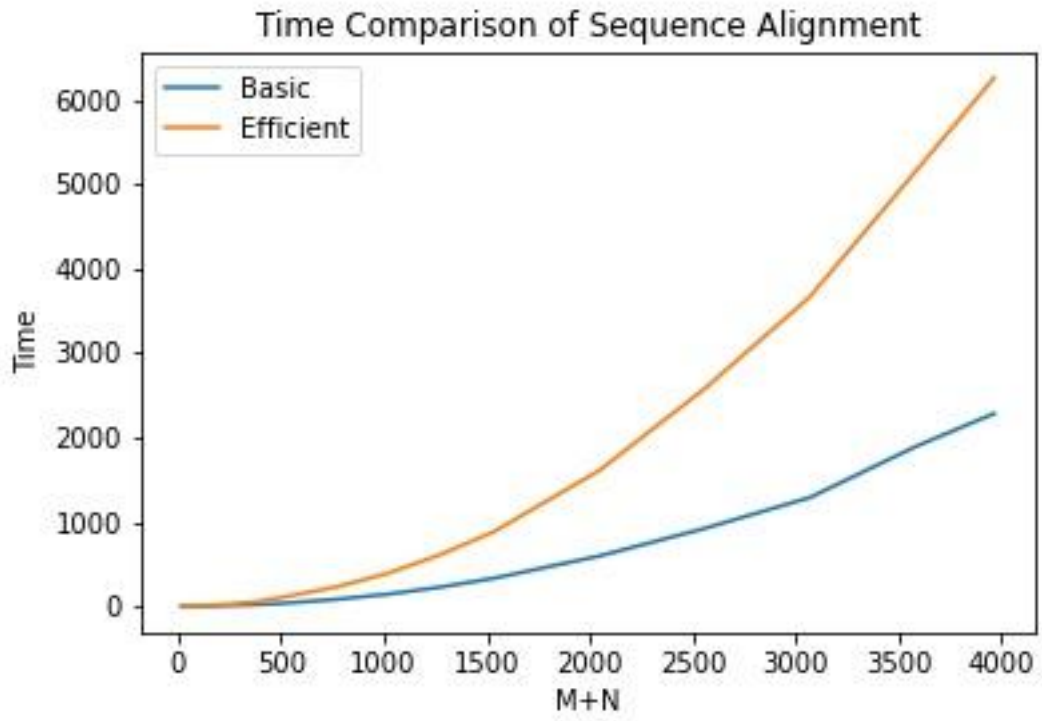
Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)

Basic: polynomial
Efficient: linear

Explanation:

- The line for the basic program is polynomial, and the line for the efficient program is linear or somewhere in between linear and polynomial. But for sure the basic program grow a lot faster than the efficient program.
- From the graph, we can see the memory for the basic program dramatically increases while the memory for efficient program increases very slowly as the problem size rises.
- This make sense because the space for efficient dynamic programming only takes O(m+n) while the space for basic programming is almost O(m*n). As input size increases, the used memory for the opt matrix of the basic program increases, and it make sense that O(m*n) grows faster than O(m+n).

Graph2 – Time vs Problem Size (M+N)



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

Basic: polynomial
Efficient: polynomial

Explanation:

- The line for the basic program is polynomial, and the line for the efficient program is also polynomial. But the efficient program grows a lot faster than the basic program.
- The graph shows that basic program takes more time compared to the efficient program as the problem size rises.
- Both programs have time complexity of O(m*n). But the constant really makes a different here. The basic program went through the 2 complete strings. But for the efficient program, on top of going through the 2 strings, more work needs to be done for the divide and conquer step. Therefore, it make sense that the efficient program line is steeper than the one for the basic 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>
3598536223: Equal Contribution
8799851341: Equal Contribution
3972192394: Equal Contribution