SUMMARY

USC ID/s: 1733072222, 5549510111, 9084470206

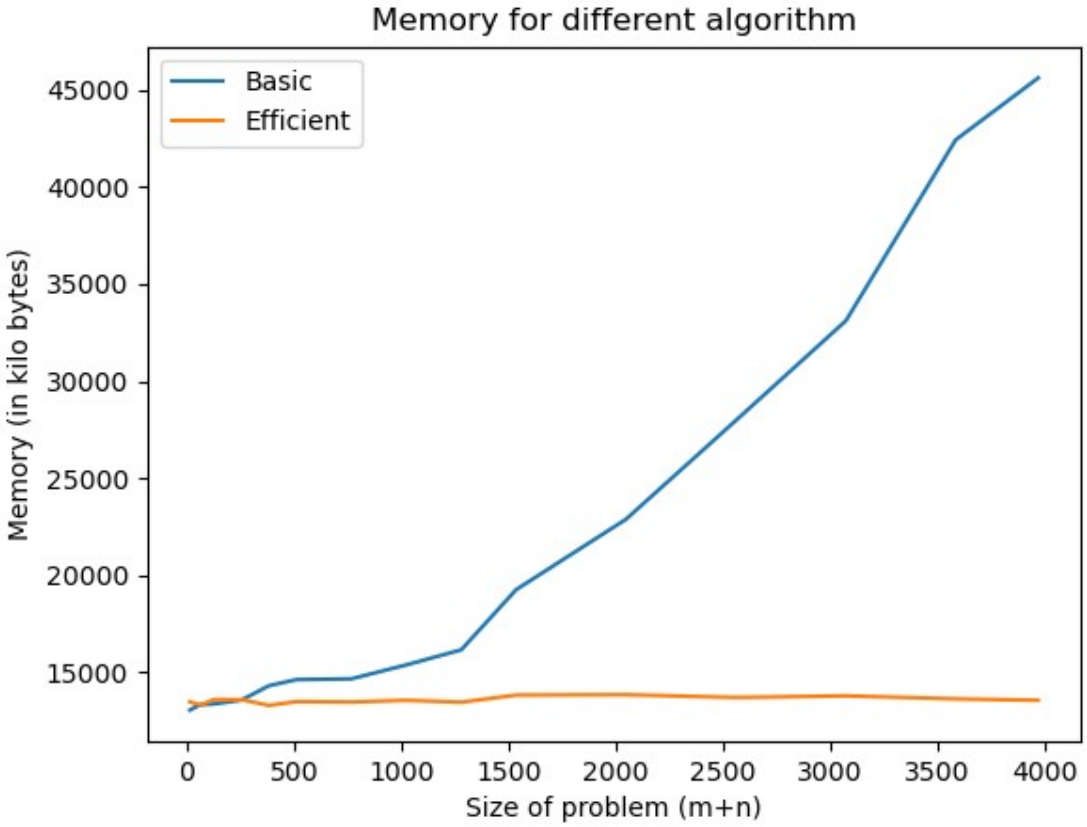
M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.04887580871582031	0.1270771026611328	13068	13492
64	0.48542022705078125	1.0993480682373047	13348	13324
128	1.9197463989257812	3.9033889770507812	13380	13608
256	7.677793502807617	13.79537582397461	13580	13604
384	15.489339828491211	30.308008193969727	14320	13308
512	27.685880661010742	55.263996124267578	14636	13504
768	61.426877975463867	124.50313568115234	14676	13480
1024	116.17064476013184	219.82693672180176	15392	13572
1280	202.62265205383301	331.03156089782715	16168	13472
1536	292.62089729309082	491.49489402770996	19268	13840
2048	568.99952888488770	876.68061256408691	22896	13860
2560	883.01944732666016	1404.2077064514160	27984	13716
3072	1343.4312343597412	2109.6804141998291	33112	13800
3584	1499.3379116058350	2812.7038478851318	42432	13644
3968	2161.1313819885254	3413.3551120758057	45616	13572

Datapoints

Insights

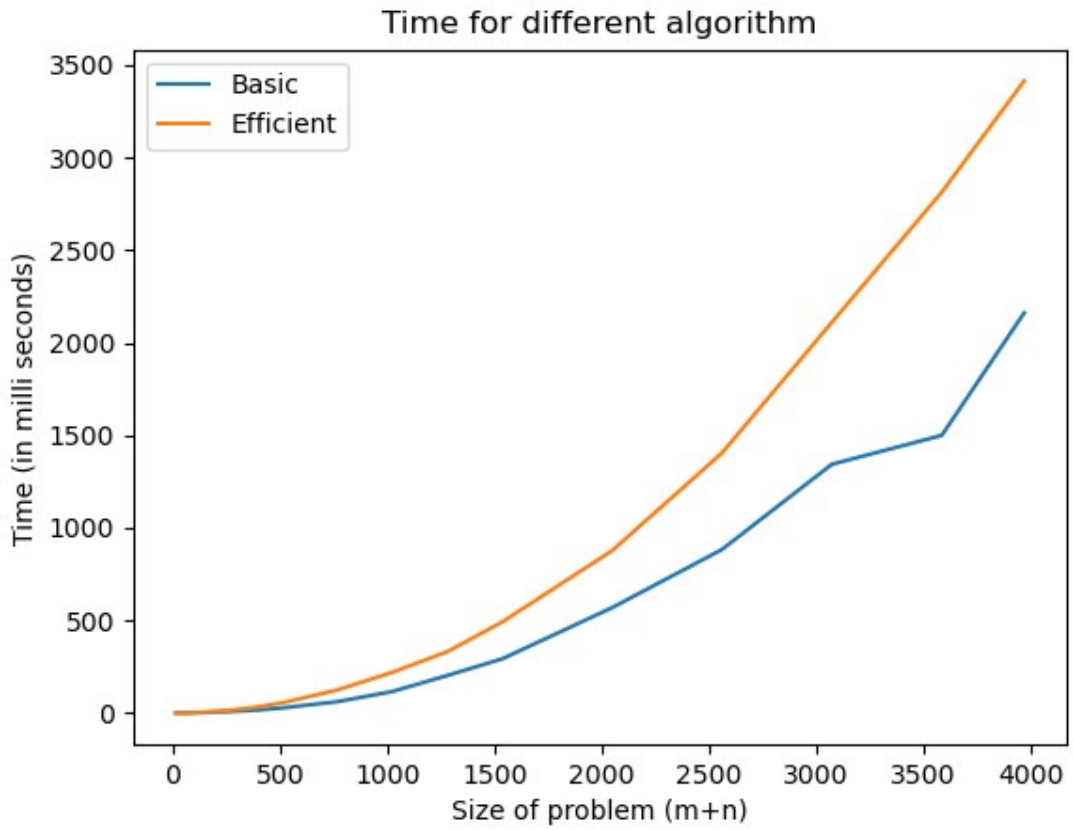
The memory differneces for the efficient vs basic can be clearly seen for bigger input sizes. Efficient takes more time to exceute than the basic though because of a bigger constant factor in the time complexity.

Graph1 – Memory vs Problem Size (M+N)



Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)
Basic: O(mn) - Polynomial (Quadratic)
Efficient: O(m+n) - Linear
Explanation: The memory requirement is much lesser in the efficient algorithm as we no longer have to save the whole 2D array and start saving columns for the Divide and Conquer algorithm. The implications can be clearly seen for larger input sizes where the efficeimt algorithm is really close to the x-axis whereas the basic algorithm starts shooting up.

Graph2 – Time vs Problem Size (M+N)



Nature of the Graph (Logarithmic/ Linear/ Polynomial/ Exponential)
Basic: O(mn) - Polynomial (Quadratic)
Efficient: O(mn) - Polynomial (Quadratic)

Explanation:
Both algorithms have the same time complexities although the constant factor for seems to be a lot higher for the efficient algorithm leading to the efficient algorithm taking more time.

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>

1733072222: Equal Contribution

5549510111 : Equal Contribution

9084470206: Equal Contribution