SUMMARY

USC ID/s:

6075279768 (Parvathi Sanjana Pericherla)

6628249966 (Keerthi Sankaralingam)

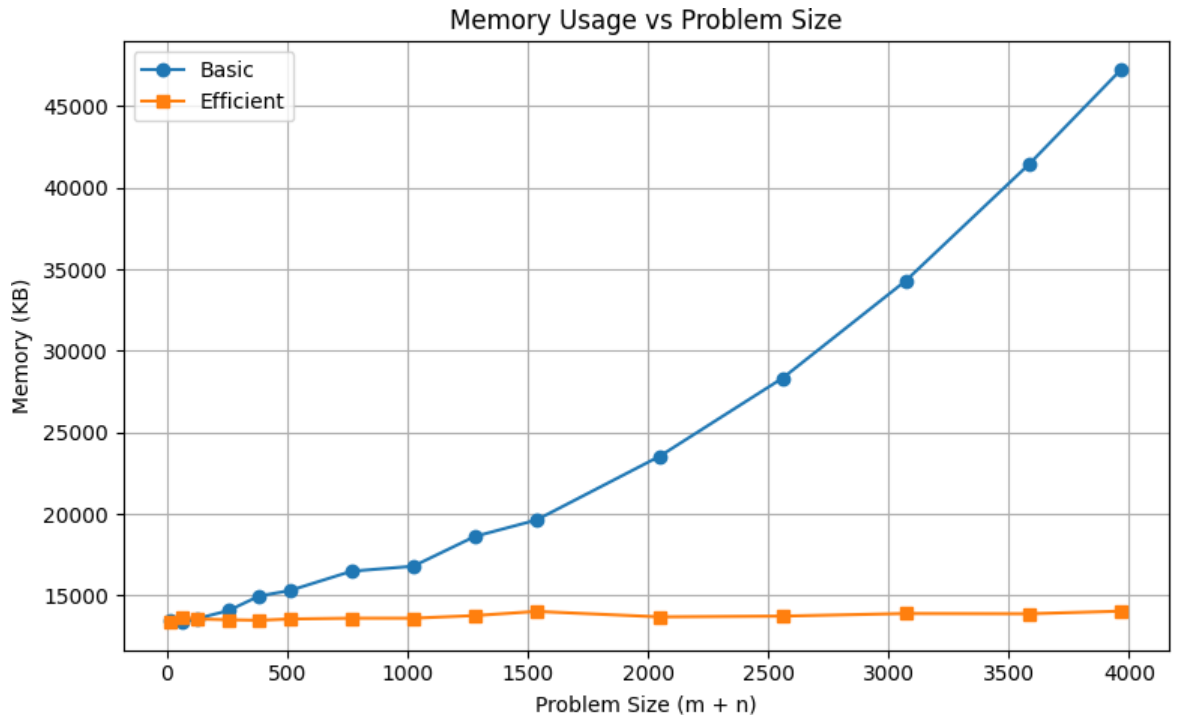
8152137082 (Shireen Chand)

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.029	0.073	14416	14544
64	0.237	0.492	14528	14448
128	0.731	1.614	14480	14352
256	3.034	5.494	14960	14672
384	7.247	11.755	16192	14432
512	11.976	20.647	16032	14336
768	29.060	45.046	15984	14432
1024	49.533	74.430	17776	14576
1280	74.956	116.817	18256	14384
1536	100.972	151.433	20544	14816
2048	191.393	283.542	23984	14736
2560	295.937	446.921	29440	14336
3072	428.844	604.230	34256	14720
3584	616.599	893.240	41440	14832
3968	778.533	1082.512	47264	14864

Insights

Graph1 – Memory vs Problem Size (M+N)



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

Basic: Polynomial
Efficient: Linear

Explanation:

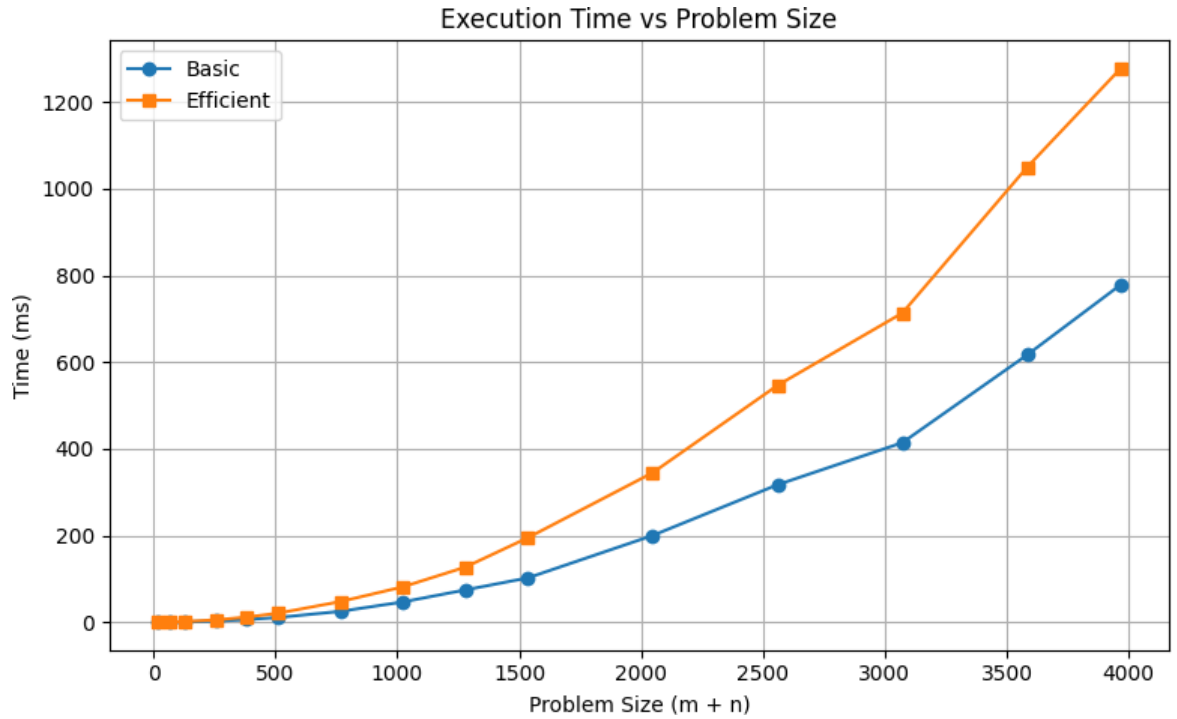
Basic Algorithm (Blue Line):

- The curve shows a significant increase in memory usage as the problem size grows.
- The sharp rise and consistent upward trend indicate a **polynomial space complexity**.
- The increase appears more dramatic for larger inputs, suggesting that the algorithm allocates memory proportionally to the square of the input size, indicating **O(m × n)** space complexity.

Efficient Algorithm (Orange Line):

- The curve remains nearly flat, indicating that the algorithm maintains a **linear** complexity, **O(n).**
- This line graph confirms that the algorithm is highly memory efficient, utilizing memory optimizations effectively.

Graph2 – Time vs Problem Size (M+N)



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

Basic: Polynomial

Efficient: Polynomial

Explanation:

Basic Algorithm (Blue Line):

- The curve shows a steady increase in execution time as the problem size increases. The increase is relatively smooth, indicating that the algorithm has a **linear or slightly polynomial complexity**.
- The slope of the curve increases consistently, suggesting that the algorithm processes each additional input size with roughly proportional additional time, indicative of **$O(n^2)$** complexity.

Efficient Algorithm (Orange Line):

- The curve initially starts off with a lower slope compared to the basic algorithm, indicating a more optimized start.
- However, as the input size increases, the slope of the curve increases sharply, suggesting a **polynomial complexity, $O(n^2)$**. This behavior indicates that while the algorithm is memory efficient, it incurs higher computational cost for larger inputs because we are recalculating the opt array.

Contribution

USC IDs:
6075279768 :Equal Contribution
6628249966: Equal Contribution
8152137082: Equal Contribution