SUMMARY

USC ID/s:

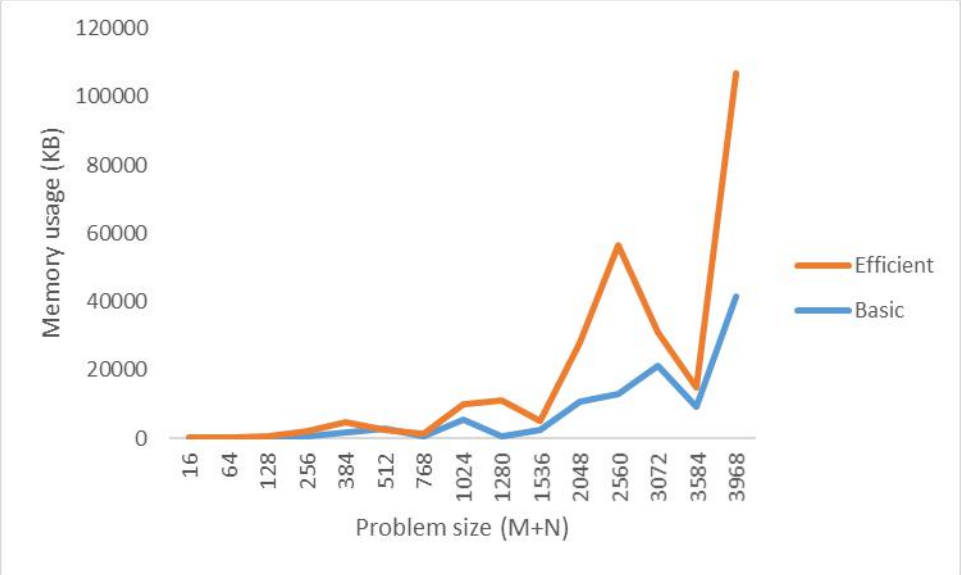
1960582960

Datapoints

M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.055	0.124	0	0
64	0.193	0.281	0	117.44
128	0.418	0.530	117.44	352.32
256	0.783	1.376	704.66	1409.31
384	1.562	2.051	1644.16	3053.49
512	2.182	3.204	2818.62	-375.16
768	3.748	4.178	595.16	773.32
1024	4.219	5.620	5494.35	4454.97
1280	5.309	8.595	370.43	10873.54
1536	6.740	11.286	2303.20	2780.63
2048	10.657	19.006	10646.48	17311.32
2560	17.194	30.896	12778.53	43584.31
3072	21.703	37.270	21102.06	10043.93
3584	31.075	54.016	9219.14	5526.69
3968	35.792	60.683	41375.34	65649.77

Insights

Graph1 – Memory vs Problem Size (M+N)



Nature of the Graph (Logarithmic/ Linear/ Exponential)

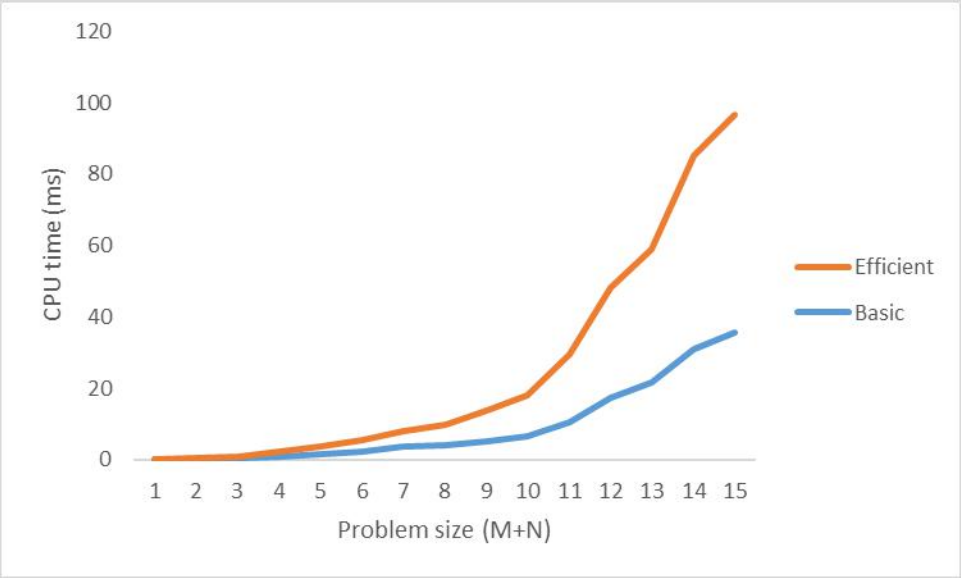
Basic: Quadratic

Efficient: Linear

Explanation:

In this graph, red line represents the DP algorithm and blue line represents the DP&DC algorithm. The space complexity of efficient algorithm which applies divide-and-conquer is O(m+n) and the basic algorithm's space complexity is O(mn). When input size is small, the difference between these two algorithms are not clear and the basic one is always smaller than the efficient one. However, when problem size increases, the memory usage of basic algorithm will grow faster than the efficient one. As we can see that the efficient version has a linear growth which leads to space optimization.

Graph2 – Time vs Problem Size (M+N)



Nature of the Graph (Logarithmic/ Linear/ Exponential)

Basic: Quadratic

Efficient: Quadratic

Explanation:

In this graph, red line represents the DP algorithm and blue line represents the DP&DC algorithm. The efficient version brings the space requirement down to linear while blowing up the running time by at most an additional constant factor. In general, Both of these two algorithm's time complexity are O(mn) and the basic version always takes less time than the efficient version.

Contribution

1960582960 : <Equal Contribution>