SUMMARY

USC ID/s: 1290146248_2212208812_8841152462

Datapoints

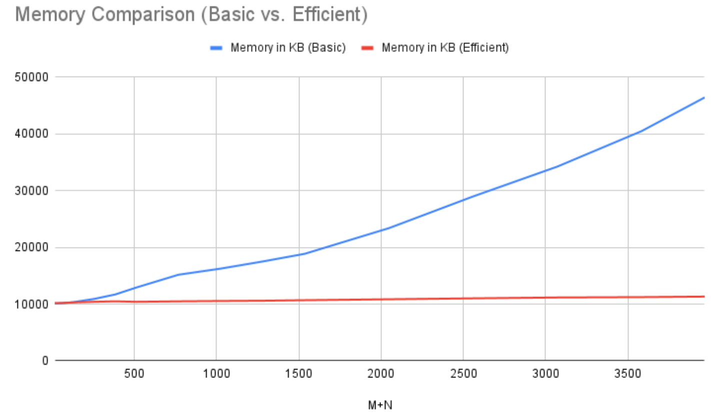
M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.51116943359375	0.6871223449707031	10124	10168
64	2.0432472229003906	3.715991973876953	10172	10204
128	3.8499832153320312	14.9688720703125	10340	10296
256	21.17180824279785	1.557083129882812	10912	10408
384	28.605222702026367	71.23899459838867	11700	10492
512	49.9567985534668	108.506202697753910316	12920	10404
768	115.96417427062988	247.01380729675293	15172	10488
1024	202.52418518066406	409.23213958740234	16248	10548
1280	315.8578872680664	684.6609115600586	17508	10596
1536	472.9418754577637	928.5581111907959	18864	10688
2048	821.1371898651123	1662.715196609497	23384	10852
2560	1343.8470363616943	2821.122169494629	28972	11016
3072	1832.1139812469482	3760.4072093963623	34232	11168
3584	2524.266004562378	5196.144104003906	40480	11228
3968	3071.820020675659	5927.416086196899	46416	11325

Insights

As the input size increase, the time and space usage increase faster in the basic version, as well as the time usage of the efficient version, but the memory usage of the efficient version seems proportional to the increase in the input size. From the datapoints we can find out that the time usage of the efficient version is always greater than (at least equal) the time usage of the basic version because we can either optimize the time or space. For the efficient version, we optimized the space usage at the expense of time usage.

Graph1 – Memory vs Problem Size (M+N)

[Add Graph1 here]



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

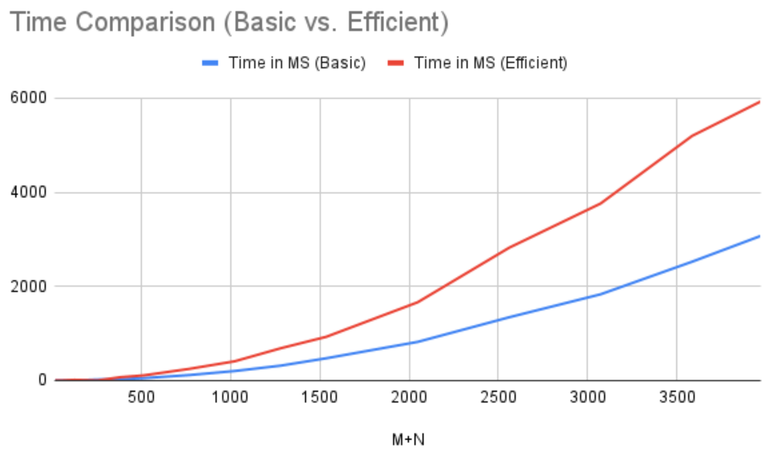
Basic: Polynomial O(mn)
Efficient: Linear O(m + n)

Explanation:

We can clearly see that the cost of memory is optimized when the problem size increases. This is because, for the recursive calls in the efficient algorithm, we work on one recursive call at a time and reuse the memory space from one call to the next. In contrast, we spend an m * n space to record all possible matching in DP.

Graph2 – Time vs Problem Size (M+N)

[Add Graph2 here]



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

Basic: Polynomial O(mn)
Efficient: Polynomial O(mn)

Explanation:

Although the times complexity of both algorithm are the same, in real world, the efficient version spend more time than the basic ones. Since we optimized the space complexity in the efficient version, we sacrifice a little bit more time (a constant factor) on recursive calls by getting a huge decrease in memory usage.

Contribution

<1290146248_2212208812_8841152462>: <Equal Contribution>