SUMMARY

USC ID/s: 1290146248_2212208812_8841152462

Datapoints

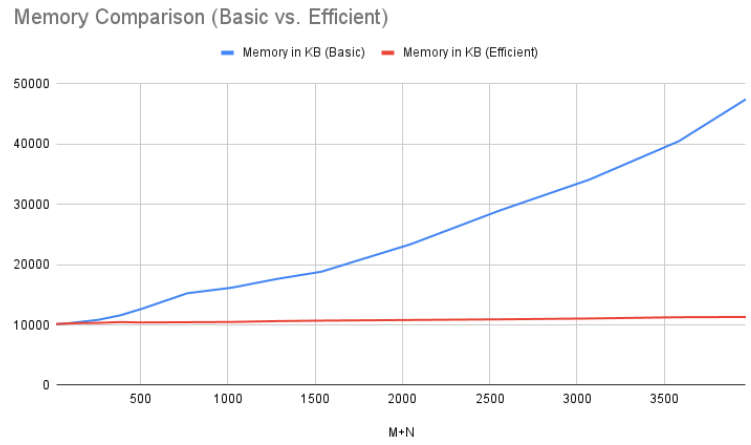
M+N	Time in MS (Basic)	Time in MS (Efficient)	Memory in KB (Basic)	Memory in KB (Efficient)
16	0.5998611450195312	0.6871223449707031	10136	10132
64	2.86102294921875	3.492116928100586	10204	10216
128	5.658864974975586	11.647939682006836	10408	10284
256	18.890857696533203	31.792879104614258	10832	10332
384	30.737876892089844	63.09986114501953	11592	10468
512	55.70030212402344	121.00100517272949	12716	10404
768	113.09123039245605	253.27181816101074	15252	10446
1024	222.2001552581787	428.9369583129883	16180	10484
1280	323.6548900604248	635.7378959655762	17616	10628
1536	475.71301460266113	906.5980911254883	18824	10708
2048	828.8929462432861	1590.142011642456	23384	10816
2560	1313.7898445129395	2483.47806930542	28996	10924
3072	1877.3260116577148	3771.138906478882	34100	11064
3584	2453.4738063812256	5527.33302116394	40464	11272
3968	3243.7970638275146	6127.490043640137	47416	11316

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>