**SUMMARY**

## USC ID/s:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M+N | Time in MS (Basic) | Time in MS (Efficient) | Memory in KB (Basic) | Memory in KB (Efficient) |
| 16 | 2.17129001021385 | 1.99169999361038 | 100.692000000000 | 100.670399999999 |
| 64 | 2.55921998620033 | 2.35508000850677 | 101.100799999999 | 100.670399999999 |
| 128 | 2.48590999841690 | 2.69299998879432 | 151.073600000000 | 151.409600000000 |
| 256 | 3.90959000587463 | 3.80607998371124 | 310.380799999999 | 360.716000000000 |
| 384 | 4.28055000305175 | 5.01157999038696 | 520.097599999999 | 629.144000000000 |
| 512 | 5.87906998395919 | 5.52884000539779 | 889.194400000000 | 1048.575200000000 |
| 768 | 7.35071998834609 | 6.96265000104904 | 1728.055200000000 | 1988.103199999990 |
| 1024 | 9.73131999373435 | 8.54398000240325 | 1283.785600000000 | 1231.506400000000 |
| 1280 | 11.43213000893590 | 10.36116999387740 | 3485.743200000000 | 4061.840000000000 |
| 1536 | 13.47890999913210 | 13.95177000761030 | 5233.762400000000 | 6577.603200000000 |
| 2048 | 18.49639999866480 | 20.62300997972480 | 6129.532000000000 | 10386.256000000000 |
| 2560 | 25.84049999713890 | 30.69225999712940 | 8160.431200000000 | 12677.880000000000 |
| 3072 | 36.04957002401350 | 39.81088998913760 | 10925.409600000000 | 6274.298400000000 |
| 3584 | 38.32191002368920 | 38.72176000475880 | 12352.302400000000 | 3755.263200000000 |
| 3968 | 51.79842001199720 | 46.27932998538010 | 17030.703200000000 | 20935.180800000000 |

## Datapoints

## Insights

### Graph1 – Memory vs Problem Size (M+N)

图表, 折线图

描述已自动生成

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

Basic: O(n^2)

Efficient: O(n)

#### Explanation:

The basic algorithm costs O(n^2) memory, while the efficient algorithm costs O(n) memory. However, at some datapoints, the efficient one costs more memory than the basic one. It probably has following reasons: (1) the minimum cost divide point is not ideal that one string is empty while the other string is very long, which costs the same memory as the basic algorithm; (2) the number of recursion call is unstable, which depends on the minimum cost divide point, and each recursion stack also costs memory; (3) the JVM memory architecture in JAVA; and (4) the operating system environment, etc. Anywhere, it’s understandable, because O(n) algorithm doesn’t guarantee that it runs faster than O(n^2) algorithm.

### Graph2 – Time vs Problem Size (M+N)

图表, 折线图

描述已自动生成

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

Basic: O(n^2)

Efficient: O(n^2)

#### Explanation:

Even though both basic algorithm and efficient algorithm costs O(n^2), the efficient one takes a little bit more time than the basic one by exchanging less memory in used than the basic one.

## 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>