**SUMMARY**

## USC ID/s: 1417356257(Qingyu Zhou)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M+N | Time in MS (Basic) | Time in MS (Efficient) | Memory in KB (Basic) | Memory in KB (Efficient) |
| 16 | 0.161 | 0.44 | 983040 | 983040 |
| 64 | 0.344 | 0.494 | 983040 | 999424 |
| 128 | 0.715 | 1.425 | 1015808 | 1015808 |
| 256 | 1.278 | 3.74 | 1212416 | 1048576 |
| 384 | 2.118 | 4.477 | 1638400 | 1048576 |
| 512 | 4.099 | 7.19 | 1998848 | 1146880 |
| 768 | 16.606 | 22.368 | 3309568 | 1163264 |
| 1024 | 20.717 | 37.086 | 5144576 | 1261568 |
| 1280 | 37.86 | 56.919 | 7520256 | 1327104 |
| 1536 | 42.748 | 57.827 | 11321344 | 1327104 |
| 2048 | 55.633 | 99.683 | 19021824 | 1458176 |
| 2560 | 105.121 | 128.192 | 28688384 | 1671168 |
| 3072 | 131.308 | 167.042 | 40648704 | 1753088 |
| 3584 | 156.693 | 213.762 | 54558720 | 1982464 |
| 3968 | 208.405 | 270.72 | 66453504 | 2031616 |

## Datapoints

## Insights

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

Chart, line chart

Description automatically generated

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

Basic: Polynomial

Efficient: Linear

#### Explanation: The basic algorithm builds a (m+n+1)^2 array storing the minimum costs, so the memory used by basic algorithm is O(n^2). The efficient algorithm only takes a 2\*(m+1) array for minimum costs, which is O(n).

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

Chart, line chart

Description automatically generated

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

Basic: Polynomial

Efficient: Polynomial

#### Explanation: Both of the basic algorithm and efficient algorithm need to calculate (n+1)\*(m+1) elements of the minimum cost table, so the time of both algorithms is O(n^2).

## 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>