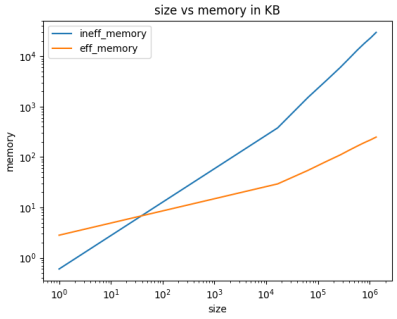
Directory Structure:

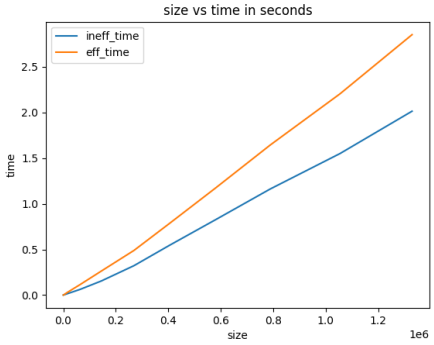
```
|
————basic.sh
|
————efficient.sh
|
————plot_data.sh
|
————final_basic.py
|
————final_efficient.py
|
————plot_data.py
|
————--main ———— efficient.py
|                       |
|                       — inefficient.py
|                       |
|                       — divide_conquer.py
|                       |
|                       — input_generator.py
|                       |
|                       — input_reader.py
|
————--drivers ———— combined_driver.py
|                       |
|                       — div_conq_driver.py
|                       |
|                       — efficient_driver.py
|                       |
|                       — inefficient_driver.py
|
————test———— plot_data_test.py
|                       |
|                       — plot_data_v2_test.py
|                       |
|                       — stress_test.py
|
————data————data_efficient.txt
|                       |
|————————————————data_inefficient.txt
```

Results:

We found that the memory efficient version on the average uses 50% lesser memory than the basic version. We observe that the primary reason for that is ensuring the number of columns is 2 in the dp array, in the space efficient version. Another factor that contributes to this is that the building of the strings happens only when the subproblems is of size less than 2.



We also found that the basic version ran quicker than the memory efficient version by a factor of less than 2x. This is because the space efficient version runs once for every divide step.



Contribution:

Input Generator:
Memory and Time plots:

Memory Inefficient Version:
Memory Efficient Version: