**Students in the group**

Akash Koratagere Shivaramu - **8016392965**

Sriharsha Sreenatha - **1357016181**

Madhura Muralidhar - **4732217257**

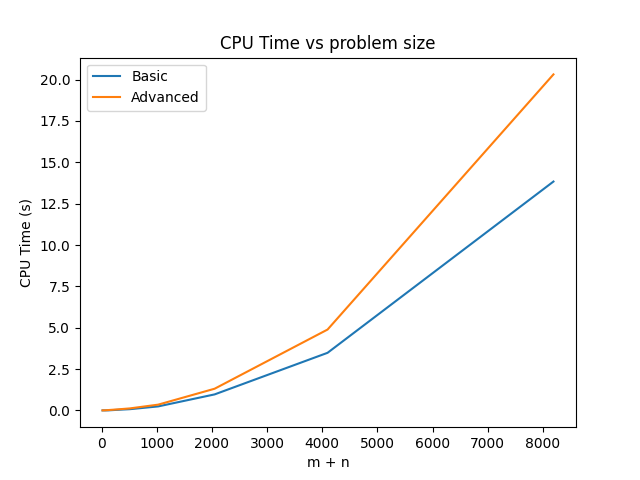
**Summary**

**Insights from CPU time vs Problem size plot**

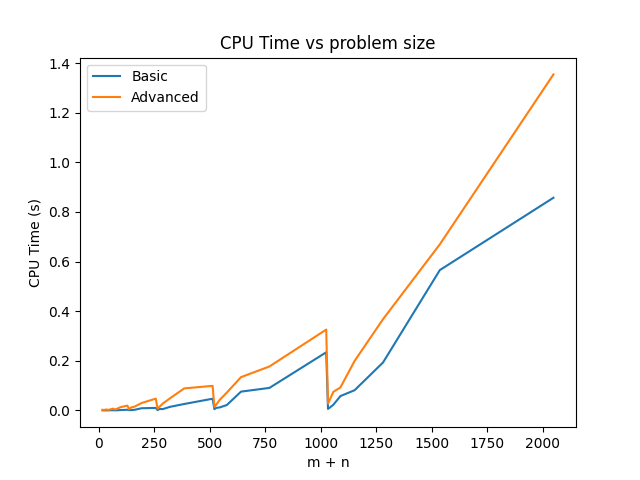
* Basic Algorithm is twice as fast as the Memory Efficient algorithm.
* This is because, it can backtrack on the memoization table efficiently rather than having to do recursive computation on sub problems to find the efficient alignment (which is the case in Memory efficient algorithm).

The following is a plot of CPU time vs problem size for both basic and advanced versions.

* The curve is ideal when m equal to n. Both strings are of same size.



* The curve has random dips when m not equal to n. Either m or n is too large compared to other.

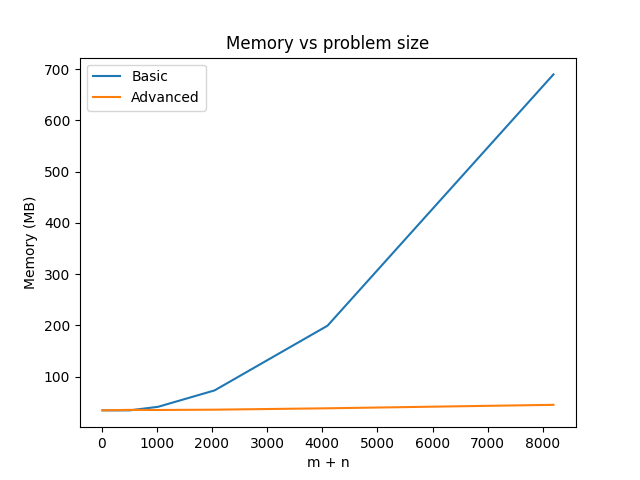


**Insights from Memory vs Problem size Plot**

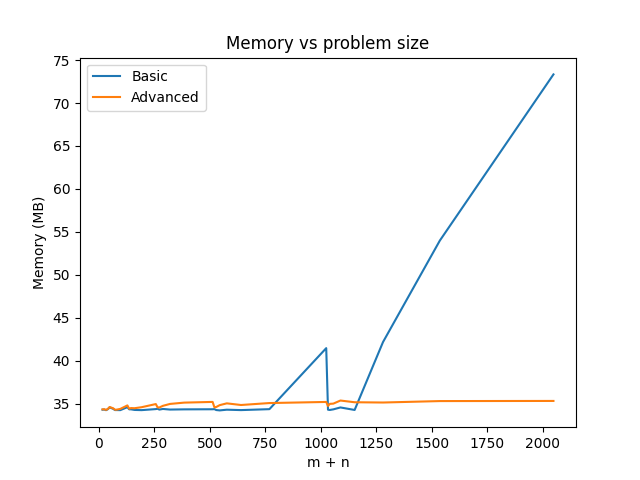
* Basic Algorithm uses a lot of memory when compared to Memory Efficient Algorithm.
* In the Basic algorithm, memory required is size m\*n for the memoization table. But the Memory efficient algorithm just requires a table of size 2\*max (m, n).
* Hence the memory requirement grows exponentially as the problem size grows in basic version as compared to linear growth in Memory efficient version.

The following is a plot of Memory taken vs problem size for both basic and advanced versions.

* The curve is ideal when m = n. Both strings are of same size.



* The curve has random dips when m != n. Either m or n is too large compared to other.



**Insights from Both Graphs**

* Basic algorithm is faster than Memory efficient algorithm.
* Memory efficient algorithm uses less memory compared to Basic algorithm when the problem size is large.

**Requirements**

For installing all the dependencies required in this project use this command

**pip install -r requirements.txt**

**Contributions of each group member**

**Akash**

* Coding of Input string generator
* Coding of Basic algorithm
* Testing of Basic algorithm
* Creation of test cases
* Plot generation
* Report creation

**Madhura**

* Coding of Basic algorithm
* Testing of Basic algorithm
* Coding of Memory efficient algorithm
* Verification of test case and insights from plots
* Report creation

**Sriharsha**

* Coding of Memory efficient algorithm
* Testing of Memory efficient algorithm
* Creation of test cases
* Verification of test case and insights from plots
* Plot generation
* Report creation