

I am working on this project alone

For this project, I create an  $O(n*m)$  graph to get my optimal cost for my basic algorithm and backtrack the same graph to get my optimal alignment. I create an  $O(n)$  graph to get optimal cost for my space efficient algorithm and use divide and conquer method to get my optimal alignment (Hirschberg's algorithm).

Before I look at my result, I expect to see an quadratic growth growth on memory and time for basic algorithm and time for efficiency algorithm since the graph I create for basic is  $O(nm)$  and Hirschberg's algorithm takes  $O(nm)$ . The result for time in both algorithm increase in quadratic time which is what I expect, an interesting to point out is for each data point, my efficient algorithm takes double amount of time compare to my basic algorithm. I assume this is cause by how I solve for optimal alignment (basic takes  $O(nm)$  time to get optimal cost and  $O(n+m)$  to get optimal alignment, efficient takes  $O(nm)$  time to get optimal cost and  $O(nm)$  to get optimal alignment). The memory plot for basic looks also like a quadratic function, although its growth rate are less then  $x^2$  (probably because of cache or how python saves memory). The growth rate of memory plot for efficient doesn't grow which surprise me, I assume this result is cause by the small input size. If we consider only the first few data points for the  $O(nm)$  time and memory costs, their growth rate also looks like 0.