**Summary**

In this project, we implement a basic version and a memory efficient version of sequence alignment. The basic version uses dynamic programming, and the memory efficient version uses dived and conquer as well as dynamic programming. Chart 1 shows the time complexity and space complexity of these two versions (m, n is the length of input string 1, 2).

|  |  |  |
| --- | --- | --- |
|  | **Time Complexity** | **Space Complexity** |
| **Basic Version** | O(m\*n) | O(m\*n) |
| **Memory Efficient Version** | O(m\*n) | O(m+n) |

**Chart 1. Complexity**

The time complexity of two versions is the same, while in experiment, we can see from Chart2 that memory efficient version runs slower than basic version. It is because that in the first level of divide and conquer, memory efficient version uses dynamic programing to get the best divide point and cost as much time as the whole of basic version. We can also find that both two time cost is a straight line, which is coincide with out time complexity.

图表, 折线图

描述已自动生成

**Chart 2. CPU Time Cost**

The space complexity of two versions is totally different. Memory efficient version costs much less than basic version. Chart 3 also proves that.

图表, 折线图

描述已自动生成 **Chart 3. Memory Cost**

**Group Members Contribution:**

Yash Prashant Naik (USCid-2948520099): coding, testing

Qishen Cai (USCid-5396727868): coding, report preparation