Question 1 (10pts)

Given two sequences *X* and *Y,* and an alphabet weight scoring matrix *s* define the following:

1. Global sequence alignment of *X* and *Y* (optimal)
2. Local sequence alignment of *X* and *Y* (optimal)

(Note: You just need to define what they are and not how to obtain them)

Question 2 (10pts)

Describe a memory-efficient algorithm for computing the local alignment between two sequences *X* and *Y*. What is the time and space complexity of your algorithm?

Question 3 (20pts)

In aligning two sequences we often want to explicitly model the gaps. Answer the following questions regarding gaps:

1. Why do we want to model gaps explicitly?
2. Describe the affine (*i.e,* linear) gap model.
3. Describe the general gap model.
4. Describe the algorithm for optimally aligning two sequences when the general gap model is used. What is the space and time complexity of your algorithm? (You need to derive the dynamic programming recurrence relations for that.)

Question 4 (20pts)

Answer the following questions regarding multiple sequence alignments:

1. Why do we use approximate algorithms for aligning multiple sequences?
2. Define the sum-of-pairs score for a multiple sequence alignment.
3. What is the center-star method for multiple sequence alignment? Is it an approximate algorithm? If yes, is there a bound on how poorly it can perform? (You do not need to prove this bound if it exists, just need to state what it is and state the assumptions associated with it).