1.

(a) Let

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + \sigma(a_i, b_j) \\ S_{i-1,j-1-p} + \sigma(a_i, b_j - p) + w(p) & p = 1, 2, ..., j-1 \\ S_{i-1-q,j-1} + \sigma(a_i - q, b_j) + w(q) & p = 1, 2, ..., i-1 \end{cases}$$

Now imagine a score $S'_{i,j}$ for the same position that is greater than $S_{i,j}$. This means that one of

$$S_{i-1,j-1} + \sigma(a_i, b_j)$$
$$S_{i-1,j-1-p} + \sigma(a_i, b_j - p)$$
$$S_{i-1-q,j-1} + \sigma(a_i - q, b_j)$$

is greater than $S_{i,j}$, which is a contradiction because $S_{i,j}$ is defined as their maximum.

(i) For the recursive definition to work, we should initialize the following values.

$$S_{0,0} = 0$$
$$S_{i,0} = w(i)$$
$$S_{0,j} = w(j)$$

(ii) In this case, we should initialize the following values.

$$S_{0,0} = S_{i,0} = S_{0,j} = 0$$

We then identify the cell in the M^{th} row or N^{th} column that has the greatest score and perform the traceback as normal, stopping when the 0^{th} row or column is reached.

(iii) We fill out a matrix N where $N_{i,j}$ is calculated by summing all possible one-step extensions of shorter alignments. We begin by initializing the following values.

$$S_{0,0} = S_{i,0} = S_{0,j} = 1$$

Then $N_{i,j}$ is calculated at each cell with the following recurrence.

$$N_{i,j} = N_{i-1,j-1} + \sum_{k=0}^{j-2} N_{i-1,k} + \sum_{k=0}^{i-2} N_{k,j-1}$$

When N is completely filled out, N_{MN} will contain the total number of possible alignments.

(b) In this case, we initialize the DP table as in a-i) but we use the following recurrence to calculate $S_{i,j}$.

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + \sigma(a_i, b_j) \\ S_{i,j-p} + w(p) \\ S_{i-q,j} + w(q) \end{cases}$$

(i) This algorithm takes $O(M \cdot N)$ runtime and space to execute.

2.

My solution to this problem can be found at `https://github.com/standage/Statistics-568/tree/master/sequence_spacer`. The `ss6568` script will generate the requested number of sequences (default number of sequences is 1000, default length for each sequence is 100) given a k^{th}-order Markov model. Model files can be written by hand (examples are included), but the `smt568` script (included for convenience) can also be used to generate a k^{th}-order model file (complete with initial state probabilities and transition probabilities) given a sequence or set of sequences.