//H0

//H1~Hd

(b) Please briefly justify the correctness of your answer. [4 marks]

The entries in the first row and first column of every matrix is initialized to 0 to ignore the unaligned beginnings of both sequences. H0 is the base case matrix, which shows local alignments without inner gaps. For denotes the entry at row i, column j, with p indel budget. To arrive at entry , there are the following ways: (1) from entry by aligning residue ai and bj , so there is either a match or mismatch. Hence, no indel happens, the indel budget remains p. (2) only applicable when it still has indel budget (p>0): from (i-1, j, p-1) by deleting a residue in the horizontal sequence (aligning ai with a gap position), thus p needs to be reduced by 1. (3) only applicable when it still has indel budget (p>0): from (i, j-1, p-1) by inserting a residue in the horizontal sequence (aligning bj with a gap position), thus p needs to be reduced by 1. If each of these three ways gives H (i, j, p) value< 0, we assign H (i, j, p) to 0. We do not care which way it takes (do not need to waste any indel budget to arrive at a 0 entry), since a 0 entry cannot be in the middle or at the end of an optimal local alignment. Tracing back is facilitated by keeping a record of the max value and using another 3d matrix to record the path while calculating the value for the entries.

(c) What is the time complexities of your solution to (a)? [2 marks]

O(mnd)

O(mn) for filling H0, ­­O(mnd) for calculating H1~Hd. Since local alignment does not necessarily form a 2d+1 band around the main diagonal, all entries in these matrices have to be calculated. The time required for calculating each entry is O(1). Hence, the total time required= O(mn\*1)+O(mnd\*1)=O(mnd)