**Algorithm 1:** Dynamic_Programming_Approach_Edit_Distance_bis

**Input:** X, Y

```
/* Parameters :                                                    */
/*    X , Y : strings                                             */
/* Return :                                                       */
/*    ed :  integer, optimal edit distance between X and Y.       */
/*    alignment :  array of instructions, to go from Y to X.      */
/*                                                                */
```

1 R ←ed_dynamic_mat(X,Y)// top-left to bottom right
2 L ←backward_ed_dynamic_mat(X,Y)// bottom right to top-left
3 S ←R+L;
4 ed ←S[0,0];
5 alignment ←extract optimal "path" from S.;

```
/* The optimal path is one of the possible path which goes from top-left to bottom-rigth cell of
   S, using only the cells with minimum number (the edit distance).                              */
```

6 **return** {ed : ed , alignment : alignment}