String "a" + "b"

Algorithm:
public String getShortestSuperSequence(String x, String y):
In out: Two strings which will be used to see common super sequence
Output: resulting String that is shortest super sequence
initialize String ← int matrixArr[][] = new int[(x.length+1)][(y.length+1)] // set 2d array of size of strings +1;
for each element in xAxis:
    for each element in yAxis: // create matrix
        check if x position is 0:
            set position (x,y) as y;
        check else if y position is 0:
            set position (x,y) as x;
        else check if x letter equals letter from y:
            set position (x,y) as 1 + diagonal of (x-1, y-1)
        else if nomatch or indel:
            set position (x,y) as minimum of diagnol, vertical, or horizontal
get index for value at end to get the score of matrix.
while xlength and ylength is not zero:
    if condition to check if diagnol position (x-1, y-1) characters equal:
        append to string the letter
        decriment xlength, ylength and index of score.
    else check if horizontal index greater:
        append the letter at position
        decriment posindex and index.
    else if vertical greater than horizontal:
        append letter at position
        decriment index & pos. index.
while x length not zero and ylength is zero:
        append all left over letters;
        decriment posindex & index.
while ylength not zero and xlength is zero
        append all left over letters.
        decriment pos index & index value.
return string;

3)

match = +1

mismatch = -1

gap open = -20 gap ext =

v \ w		G	G	A	C	G	T	A	C	G
	0	-21	-22	-23	-24	-25	-26	-27	-28	-29
T	-21	-1	-22	-23	-24	-25	-24	-27	-28	-29
A	-22	-22	-2	-21	-24	-25	-26	-23	-28	-29
C	-23	-23	-3	-22	-20	-25	-26	-27	-22	-29
G	-24	-24	-4	-23	-21	-19	-26	-27	-28	-21
G	-25	-25	-5	-24	-22	-20	-20	-27	-28	-27
G	-26	-26	-6	-25	-23	-21	-21	-21	-28	-27
T	-27	-27	-25	-26	-24	-22	-22	-22	-23	-28
A	-28	-28	-26	-27	-25	-23	-23	-23	-24	-23
T	-29	-29	-27	-28	-26	-24	-24	-24	-25	-24

v = TACGGGTAT

w = GGACGTACG

sequence = G

score = -7

1)

match = +1
mismatch = -1
indel/del = -1

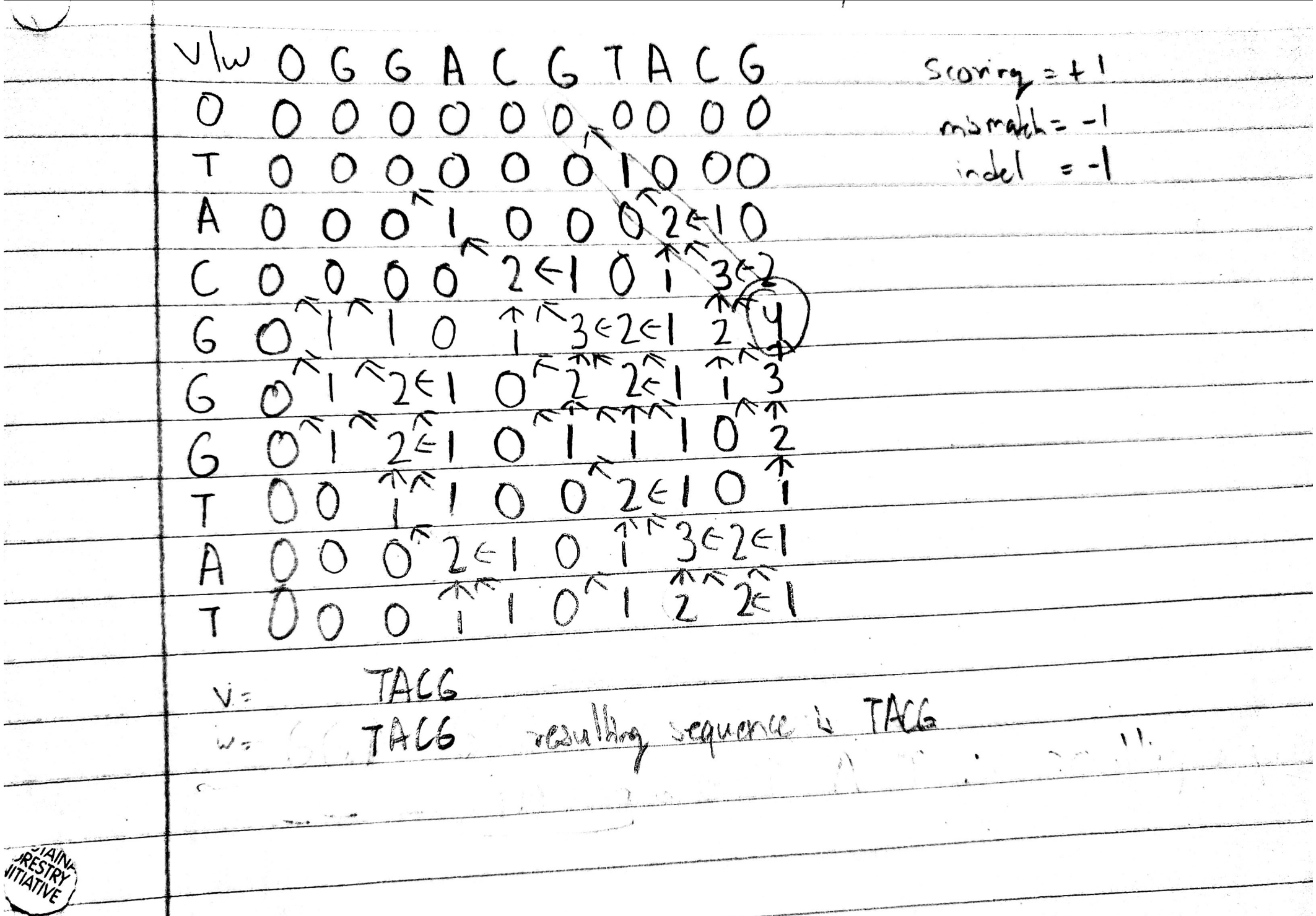
v\w		0	G	G	A	C	G	T	A	C	G
	0	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
T	-1	-1	-2	-3	-4	-5	-6	-5	-6	-7	-8
A	-2	-2	-3	-4	-3	-4	-5	-6	-5	-6	-7
C	-3	-3	-4	-5	-4	-3	-4	-5	-6	-5	-6
G	-4	-4	-3	-4	-5	-4	-3	-4	-5	-6	-5
G	-5	-5	-4	-3	-4	-5	-4	-5	-6	-5	-6
G	-6	-6	-5	-4	-5	-6	-5	-6	-7	-6	-7
T	-7	-7	-6	-5	-6	-7	-6	-5	-6	-7	-8
A	-8	-8	-7	-6	-5	-6	-7	-6	-5	-6	-7
T	-9	-9	-8	-7	-6	-7	-8	-7	-6	-7	-8

score of optimal global alignment = -1

v = _ T A C G G G T A _ T

w = G G A C G _ - T A C G

sequence = ACGTA


v/w O G G A C G T A C G
O 0 0 0 0 0 0 0 0 0 0
T 0 0 0 0 0 0 0 1 0 0
A 0 0 0 1 0 0 0 2 1 0
C 0 0 0 0 2 1 0 1 3 2
G 0 1 1 0 1 3 2 1 2 4
G 0 1 2 1 0 2 2 1 1 3
G 0 1 2 1 0 1 1 1 0 2
T 0 0 1 1 0 0 2 1 0 1
A 0 0 0 2 1 0 1 3 2 1
T 0 0 0 1 1 0 1 2 2 1
Scoring = +1
mismatch = -1
indel = -1
v = TACG
w = TACG
resulting sequence is TACG