Restore the LCS

Assuming that for some $1 < i, j$ we have $M(i,j) = r$

Observ the submatrix $\begin{array}{c|cc} & j-1 & j \\ i-1 & r_1 & r_2 \\ i & r_3 & r \end{array} \subseteq M$.

If $r \neq r_2$ and $r \neq r_3$: then

from the way we built M, we know that $a_i = b_j$ and we should keep traverse M from entry $(i-1, j-1)$, not before we keep a_{i-1} or b_{j-1} to record the LCS

else $r = r_2$ or $r = r_3$,

If $r = r_2$ then from the way we built M we can conclude that we should keep traverse M from entry $(i-1, j)$,

else we should keep traverse M from entry $(i, j-1)$.

Frome all the above, it's easy to see how to restore the LCS.

restoreLCS(M[n+1][m+1], a_n) /* could use bm instead */

0. do $k \leftarrow LCS(a_n, b_m)$, $i \leftarrow n+1$, $j \leftarrow m+1$
1. init list L.
2. while $1 < k$ do

    if $M(i, j) = M(i-1, j)$ then do

        $i \leftarrow i-1$

    else if $M(i, j) = M(i, j-1)$ then do

        $j \leftarrow j-1$

    else do /* $M(i,j) \in LCS$ */

        $i \leftarrow i-1$

        $j \leftarrow j-1$

        $k \leftarrow k-1$

        L.addToHead(A[i-1]) /* could instead add B[j-1] */

3. print L.