1. DP, $f[i][j]$ denote whether $s1[1 \dots i]$ and $s2[1 \dots j]$ can interleave to get $s3[1 \dots i+j]$.
$f[i][j] = (s1[i] == s3[i+j])\&\&f[i-1][j]||(s2[j] == s3[i+j])\&\&f[i][j-1]$. $O(n^2)$.

this DP is similar to the DP for LCS, so some advanced algorithms for LCS can be applied.
2. $O(n^2/\log n)$ by method of four russians. divide the $n \times n$ DP matrix into blocks of size $t \times t$, where $t = O(\log n)$. (for LCS: [2])
3. $O(n^2/w)$ by bit packing. (for LCS: [1])

# References


[1] Maxime Crochemore, Costas S Iliopoulos, Yoan J Pinzon, and James F Reid. A fast and practical bit-vector algorithm for the longest common subsequence problem. *Information Processing Letters*, 80(6):279–285, 2001.

[2] William J Masek and Michael S Paterson. A faster algorithm computing string edit distances. *Journal of Computer and System sciences*, 20(1):18–31, 1980.