1. Recurrence
2. Memoize
3. Order
4. Analysis
5. Optimize

- Redundant cases

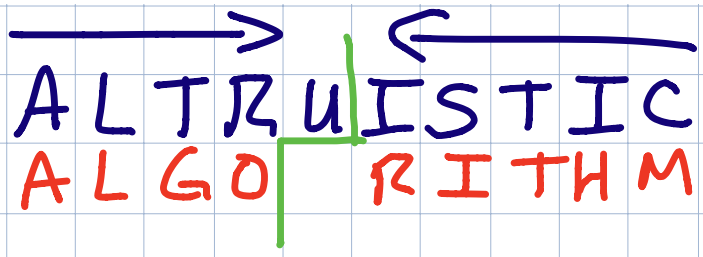
- Space
  if we only want cost
  but what if we want
  the optimal structure?

→ Edit(A[1..i], B[1..j])

$$Edit(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \left\{ \begin{array}{l} Edit(i-1,j)+1, \\ Edit(i,j-1)+1, \\ Edit(i-1,j-1)+[A[i] \neq B[j]] \end{array} \right. & \text{otherwise} \end{cases}$$

Given the entire memo table, we can reconstruct opt. seq.

Edit		A	L	G	O	R	I	T	H	M
	0	1	2	3	4	5	6	7	8	9
A	1	**0**	1	2	3	4	5	6	7	8
L	2	1	**0**	1	2	3	4	5	6	7
T	3	2	1	1	2	3	4	**4**	5	6
R	4	3	2	2	2	**2**	3	4	5	6
U	5	4	3	3	3	3	3	4	5	6
I	6	5	4	4	4	4	**3**	4	5	6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	**4**	5	6
I	9	8	7	7	7	7	**6**	5	5	6
C	10	9	8	8	8	8	7	6	6	6



Which prefix of B aligns with first half of A?

$O(mn)$ time, $O(n)$ space

$PrefLeft(i) = Edit(A[1..m/2], B[1..i])$

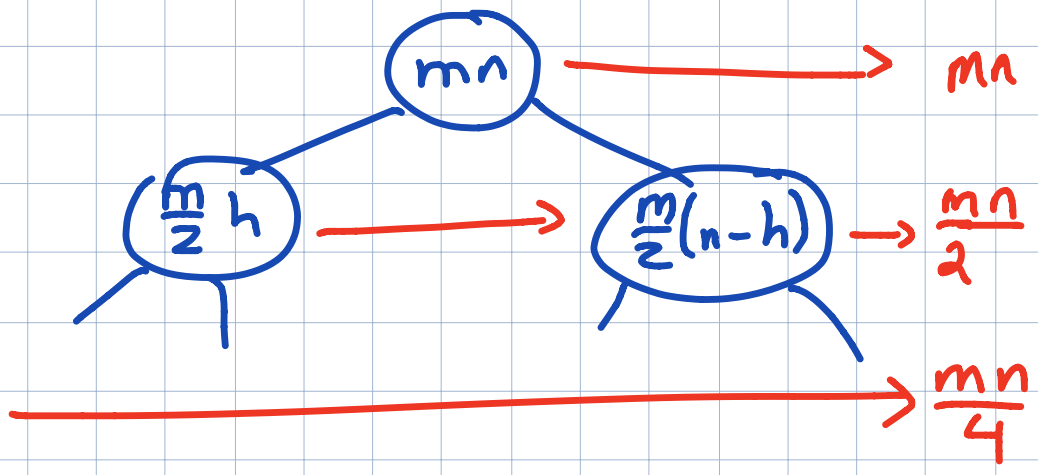
$SuffRight(i) = Edit(A[m/2+1..m], B[i..n])$

$Edit(A,B) = \min_i \left(PrefLeft(i) + SuffRight(i+1) \right)$

Edit		A	L	G	O	R	I	T	H	M
	0	1	2	3	4	5	6	7	8	9
A	1	**0**	1	2	3	4	5	6	7	8
L	2	1	**0**	1	2	3	4	5	6	7
T	3	2	1	1	2	3	4	**4**	5	6
R	4	3	2	2	2	**2**	3	4	5	6
U	5	4	3	3	3	3	3	4	5	6
I	6	5	4	4	4	4	**3**	4	5	6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	**4**	5	6
I	9	8	7	7	7	7	**6**	5	5	6
C	10	9	8	8	8	8	7	6	6	6

$$T(m,n) = O(mn) + T\left(\frac{m}{2}, h\right) + T\left(\frac{m}{2}, n-h\right)$$

$$T(0,n) = O(n)$$
$$T(m,0) = O(m)$$



$$S(m,n) = \max \begin{cases} O(n) \\ S(\frac{m}{2}, h) + S(\frac{m}{2}, n-h) \end{cases}$$

$S(\frac{m}{2}, h) = O(h)$

$S(\frac{m}{2}, n-h) = O(n-h)$

Time $= O(mn)$ — $2mn$ writes

Space $= O(n)$ — $3n$ cells

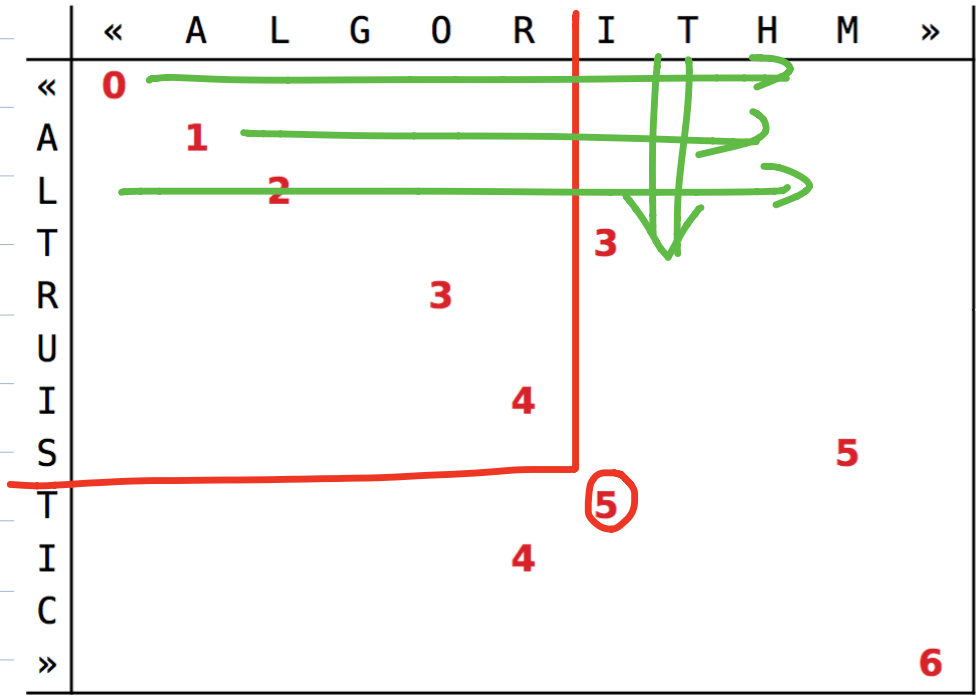
Edit		A	L	G	O	R	I	T	H	M
	0	1	2	3	4	5	6	7	8	9
A	1	**0**	1	2	3	4	5	6	7	8
L	2	1	**0**	1	2	3	4	5	6	7
T	3	2	1	1	2	3	4	**4**	5	6
R	4	3	2	2	2	**2**	3	4	5	6
U	5	4	3	3	3	3	3	4	5	6
I	6	5	4	4	4	4	**3**	4	5	6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	**4**	5	6
I	9	8	7	7	7	7	**6**	5	5	6
C	10	9	8	8	8	8	7	6	6	6

Half		A	L	G	O	R	I	T	H	M
	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
A	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
T	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
R	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
U	0	1	2	3	4	5	6	7	8	9
I	0	1	2	3	4	5	**5**	5	5	5
S	0	1	2	3	4	5	5	5	5	5
T	0	1	2	3	4	5	5	**5**	5	5
I	0	1	2	3	4	5	**5**	5	5	5
C	0	1	2	3	4	5	5	5	5	5

LCS(i,j) = length of longest common subsequence of A[1..i] and B[1..j]

$$LCS(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ LCS(i-1, j-1) + 1 & \text{if } A[i] = B[j] \\ \max\{LCS(i, j-1),\ LCS(i-1, j)\} & \text{otherwise} \end{cases}$$

	«	A	L	G	O	R	I	T	H	M	»
«	**0**	0	0	0	0	0	0	0	0	0	0
A	0	**1**	1	1	1	1	1	1	1	1	1
L	0	1	**2**	2	2	2	2	2	2	2	2
T	0	1	2	2	2	2	**3**	3	3	3	3
R	0	1	2	2	**3**	3	3	3	3	3	3
U	0	1	2	2	3	3	3	3	3	3	3
I	0	1	2	2	3	**4**	4	4	4	4	4
S	0	1	2	2	3	4	4	4	4	**5**	5
T	0	1	2	2	3	4	**5**	5	5	5	5
I	0	1	2	2	3	**4**	5	5	5	5	5
C	0	1	2	2	3	4	5	5	5	5	5
»	0	1	2	2	3	4	5	5	5	5	**6**



1. Find all matching pairs (interesting places in table) ← O(m+n) +K

2. Consider matches in the right order

Value@match = 1+max Value

$$LCS(i,j) = \begin{cases} 0 & \text{if } i = j = 0 \\ \max\{LCS(i',j') \mid A[i'] = B[j'] \text{ and } i' < i \text{ and } j' < j\} + 1 & \text{if } A[i] = B[j] \\ \max\{LCS(i',j') \mid A[i'] = B[j'] \text{ and } i' \le i \text{ and } j' \le j\} & \text{otherwise} \end{cases}$$

```
SPARSELCS(A[1..m], B[1..n]):
  Match[1..K] ← FINDMATCHES(A, B)
  Match[K+1] ← (m+1, n+1)        ⟨⟨Add end sentinel⟩⟩
  Sort M lexicographically
  for k ← 1 to K
      (i, j) ← Match[k]
      LCS[k] ← 1                  ⟨⟨From start sentinel⟩⟩
      for ℓ ← 1 to k−1
          (i', j') ← Match[ℓ]
          if i' < i and j' < j
              LCS[k] ← min{LCS[k], 1 + LCS[ℓ]}
  return LCS[K+1] − 1
```

$O(m+n+K+K\log K+K^2)$ time

NeedlemanWunch '73 → KlogK