DP:

- Longest inc. subseq.: use table $T[0 \dots n] \Rightarrow T[i]$ last el. of LIS w/ length i.

  init.: $T[0] \leftarrow -\infty$, $T[1 \cdots n] \leftarrow \infty$

  comp.: for each a_k get LIS extendable w/ a_k. since T sorted $\Rightarrow$ get pos. w/ binary search: pos l of rightest element smaller than a_k.

  $\Rightarrow T[l] < a_k < T[l+1]$. then $T[l+1] \leftarrow a_k$.

  execute in inc. order.

  get sol.: n where $T[n] \neq \infty$, $T[n+1] = \infty$.

  ~~1~~, ~~4~~, ~~6~~, ~~2~~, ~~10~~, ~~8~~, ~~12~~

  1 ⊖ ~~4~~ ⊖ ~~6~~ ⊖ 10 ⊖ 12

  2 – 5

- Longest comm. subseq. of $A = (a_1, \cdots, a_m)$, $B = (b_1, \cdots, b_n)$

  use table $T[0 \cdots m][0 \cdots n] \Rightarrow T[i][j]$ length of LCS $(a_1, \cdots, a_i), (b_1, \cdots, b_j)$

  init. $L[i,0] \leftarrow 0$, $L[0,j] \leftarrow 0$ (no init. in java then).

  comp.: $L[i,j] \leftarrow \max(L[i-1,j-1] + \underbrace{\delta_{ij}}_{\substack{1 \text{ if } a_i = b_j \\ 0 \text{ else}}}, L[i,j-1], L[i-1,j])$

  LCS is just one of $\underbrace{(a_1, \cdots, a_{i-1}), (b_1, \cdots, b_{j-1})}$, $\underbrace{(a_1, \cdots, a_i), (b_1, \cdots, b_{j-1})}$, $\underbrace{(a_1, \cdots, a_{i-1}), (b_1, \cdots, b_j)}$

  execute row ↓, col →.

  get sol.: $L[m][n]$.

		Z	I	E	G	E
	0	0	0	0	0	0
T	0	0	0	0	0	0
I	0	0	1	1	1	1
G	0	0	1	1	2	2
E	0	0	1	2	2	3
R	0	0	1	2	2	3

- Min. edit. distance: ∘ insert char, delete char, change char

  Use table $E[0 \dots m][0 \dots n] \Rightarrow E[i][j]$: MID $(a_1, \cdots, a_i) \to (b_1, \cdots, b_j)$.

  init.: $E[i][0] \leftarrow i \cdot w_{in}$, $E[0][j] \leftarrow j \cdot w_{in}$ (insert chars)

  comp. $E[i][j] \leftarrow \min(\underbrace{E[i-1][j] + w_{del}}_{\text{delete } i_{th} \text{ char } A}, \underbrace{E[i][j-1] + w_{ins}}_{\text{add char } B}, \underbrace{E[i-1][j-1] + \delta_{ij}}_{\substack{\text{change } a_i \text{ to } b_j, \\ \text{if } a_i = b_j \ \delta_{ij} = 0 \\ \text{else } \delta_{ij} = w_{ch}}})$

  $(w_{in} = w_{del} = w_{ins} = 1)$

  execute row ↓, col →

		Z	I	E	G	E
	0	1	2	3	4	5
T	1	1	2	3	4	5
I	2	2	1	2	3	4
G	3	3	2	2	2	3
E	4	4	3	2	3	2
R	5	5	4	3	3	3

- Matrixchain: $A_1 \times \cdots \times A_n$ => best brackets

  Use table $M[1\cdots n][1\cdots n]$ => $M[p][q]$ min. cost of $A_p \times \cdots \times A_q$.

  init.: $M[p][p] \leftarrow 0$, only interested in upper diag.

  comp.: $M[p][q] \leftarrow \min_{p \le i \le q} \Big(\underbrace{M[p][i]}_{\text{cost first part}} + \underbrace{M[i+1][q]}_{\text{cost second part}} + \underbrace{\text{cost } (A_p \times \cdots \times A_i) \otimes (A_{i+1} \times \cdots \times A_q)}_{\text{cost to comb.: } A^{m \times n} \times B^{n \times p} : m \cdot n \cdot p} \Big)$

  execute 

  $A_1^{k\times 1}, A_2^{1\times k}, A_3^{k\times 1}$

	1	2	3
1	0	k^2	$2k$
2		0	k
3			0

- Subset sum: Check if $\exists z \left(\sum_{i \in I} a_i = z \right)$

  Use table $T[1\cdots n][1\cdots z]$ => $T[i][j]$: I exists s.t. $\sum_{i \in I \subseteq \{1,\dots,i\}} a_i = j$ 2

  Init.: $T[0][0] \leftarrow$ true, $T[0][s] \leftarrow$ false

  comp.: $T[i][j] \leftarrow \begin{cases} T[i-1][j] & \text{if } j < a_i \quad \text{(can't even take } a_i) \\ T[i-1][j] \text{ or } T[i-1][s-a_i] & \text{if } j \ge a_i \quad \text{(check o.o.B!)} \end{cases}$

  execute row↓ col→

  get solution: $T[n][z]$.

	0	1	2	3	4	5	6	7	8	9
–	1	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	1	0	0	0	0
3	1	0	0	1	0	1	0	0	1	0
7	1	0	0	1	0	1	0	1	1	0
3	1	0	0	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1

- Knapsack: max of values & ≤ weight.

  Use table $T[1\cdots n][1\cdots w]$ => $T[i][j]$: maxValue w/ $v_1,\dots,v_i$ & $w=j$.

  init.: $T[0, w] \leftarrow 0$.

  Compute: $T[i][j] = \max\big(\underbrace{T[i-1][j]}_{\text{don't take } i_{th}}, \underbrace{T[i-1][w-w_i] + v_i}_{\text{take } i_{th}} \big)$

  execute: row↓, col→

  => also poss. other way around, $T[i][j]$: min W. of val. $\ge j$ w/ $v_1,\dots,v_i$

  => pseudo pol.! $O(nV)$ or $O(nW)$

  Polynom: Approx! use $w_i, \left\lfloor \frac{v_i}{k} \right\rfloor, W$ instead of w_i, v_i, W.

  $\Rightarrow O(n\bar{V}) \le O\left(n^2 \frac{V_{max}}{k}\right)$. $k = \frac{\varepsilon}{n} V_{max} \Rightarrow O\left(n^{\frac{3}{\varepsilon}}\right)$. $opt \ge (1-\varepsilon)\overline{opt}$