10. 01. 2024.

CURS 13 (amf.)

METODA PROGRAMĂRII DINAMICE ♡

R. Bellman ≃50

Condiții:

C1: Condiția de subproblemă / substructură optimală = pb. se poate descompune în mai multe subp., iar sol. sa (optim global) se obț. combinând sol. sub. sale (optime locale).

C2: Condiția de suprapozabilitate: (sol.) subpb. se suprapun
↓
MEMOIZARE = str. de date adecvată în care mem. sol. sub. pt. a evita apelurile inutile recursive

1. Suma maximă într-un triunghi de numere

10
1 7

10
-7 3
900 8 -15
-2 9 20 10
↳ sau 2000

$T_{i,j}$ → $T_{i+1,j+0}$, $T_{i+1,j+1}$

Var. Greedy: eșuează cu 900 sau 2000

Var. Backtraking: 0,0,0,...0 ← ce trasee ar parcurge Backul
0,0,0...1
0,0,0...1,0
0,0,0...1,1
....
1,1,1,...1

n-1 valori binare = 2^{n-1} șiruri ! ca la ASC

000
001
010
011
100
101
111
} $2^3 = 8$ V

Varianta D&I:

$$smax(T, i, j) = \begin{cases} T[i][j] + maxim\{smax(T, i+1, j), smax(T, i+1, j+1)\} & \text{dacă } 0 \le i = n-2 \\ T[i][j], & \text{dacă } i = n-1 \end{cases}$$

↓ memoizare
smax[i][j]

Subpb. se suprapun

10
-7 3
9 8 -15
-2 9 20 10

face fix ce face și backtrackingul deci 2^{n-1}

$smax[i][j]$ = —"— ideal cu $Smax[i+1][j]$ și $smax[i+1][j+1]$

$$T = \begin{pmatrix} 10 & & & \\ -7 & 3 & & \\ 9 & 8 & -15 & \\ -2 & 9 & 20 & 10 \end{pmatrix} \qquad smax = \begin{pmatrix} 41 \rightarrow \text{soluția} & & & \\ +21 & 31 & & \\ 18 & 28 & +5 & \\ -2 & 9 & 20 & 10 \end{pmatrix}$$

→ traseul e de sus în jos urmând maxima

Complexitate: $O(n^2)$ (redetic $\frac{n^2}{2}$)

```
for j in range(n):
    smax[n-1][j] = T[n-1][j]
for i in range(n-2, -1, -1):
    for j in range(i+1):
        smax[i][j] = T[i][j] + max(smax[i+1][j], smax[i+1][j+1])
```

$smax[i][j]$ = suma maximă pe un traseu care se termină cu $T[i][j]$

$$\begin{pmatrix} 10 & & & \\ 3 & 13 & & \\ 12 & 21 & -2 & \\ 10 & 30 & \boxed{41} & 8 \end{pmatrix}$$

← depinde de cele de <u>înapoi</u>

↳ soluția

$smax[i][j] = T[i][j] + \underbrace{smax(i+1, j), smax(i+1, j+1)}_{\text{înainte}}$ depinde de cele de înainte

2. Subșirul crescător maximal

$L = [5, 9, \underline{1}, 7, \underline{3}, \underline{5}, 10, \underline{8}, \underline{12}, 4, 6]$

$lmax[i]$ = lungimea maximă a unui subșir crescător care începe cu $L[i]$

$$lmax[i] = \begin{cases} 1 + \max(lmax[j]), & L[j] \geq L[i], \quad 0 \leq i \leq n-2 \\ & i+1 \leq j \leq n-1 \\ 1, & i = n-1 \end{cases}$$

~~Seb.~~

0 1 2 3 4 5 6 7 8 9 10
smax = [4, 3, 5, 3, 4, 3, 2, 2, 1, 2, 1] ← sol. este maximul din listă

succesori = [..., 3, 4, 5, 6, 8, 8, -1, 10, -1]

mă duc pe poz. pe care le am ..