ALGORITMUSOK TERVEZÉSE ÉS ELEMZÉSE

4. Gyakorlat

DINAMIKUS PROGRAMOZÁS

- Dynamic programming (**DP**)
  - Richard Bellman
- Általában optimalizálási feladatoknál
- Részfeladatokra osztás
  - A részfeladatok nem függetlenek
    - Ilyenkor a D&Q stratégia többet dolgozna
- „programozás”: táblázatos módszer
  - A részfeladatok eredményeit táblázatban tároljuk
- Általában a feladatoknak sokféle megoldása lehet
  - Ebből 1 optimálist keresünk

DP ALGORITMUS SZERKEZETE

- 4 fő lépés
  1) **Jellemezzük az optimális megoldás szerkezetét**
  2) **Rekurzív módon definiáljuk az optimális megoldás értékét**
  3) **Kiszámítjuk az optimális megoldás értékét alulról felfelé történő módon**
  4) **A kiszámított információk alapján megszerkesztünk egy optimális megoldást**
- A 4. lépés elhagyható, ha csak az optimális értéket keressük
- A 4. lépéshez 3. lépés során kiegészítő információkat tárolunk

DP – MIKOR ÉRDEMES HASZNÁLNI??

- Ha a megoldandó feladat **optimálási részstruktúra tulajdonságú**
- Az optimális megoldás a részfeladatok optimális megoldásait is tartalmazza

ISMÉTLÉS: LEGRÖVIDEBB UTAK MINDEN CSÚCSPÁRRA

- Vizsgáljuk meg az eddig látott módszereket
- Dijkstra algoritmus
  - Futtassuk a Dijkstra algoritmust a gráf összes csúcsára, mint startcsúcsra
  - Ritka gráf esetén: $O(n * (n + m) * \log n)$ = **$O(n^2 * \log n)$**
  - Sűrű gráfok esetén: $O(n * (n + n^2) * \log n)$ = **$O(n^3 \log n)$**
- Sor alapú Bellman-Ford algoritmus
  - Futtassuk a sor alapú BF algoritmust a gráf összes csúcsára, mint startcsúcsra
  - Ritka gráf esetén: $O(n * n * m)$ = **$O(n^3)$**
  - Sűrű gráfokra: **$O(n^4)$**

FLOYD-WARSHALL ALGORITMUS

- Csúcsmátrixos reprezentációval használjuk
- D és π mátrixok
- D_{ij} az i és j csúcs közti legrövidebb út hosszát tartalmazza
- π_{ij} az i és j közti legrövidebb úton j megelőzője
- Belső csúcs:
  - Egy $p = <v_1, v_2, \ldots, v_k>$ út belső csúcsa minden v_1-től és v_k-tól különböző csúcs
- n lépésben határozza meg a legrövidebb utakat
  - A k. lépésben $D_{ij}^{(k)}$ i és j közti optimális út hosszát tartalmazza
  - Úgy, hogy a belső csúcsok címkéje legfeljebb k
- Negatív összsúlyú kör esetén a D mátrix főátlójában negatív értékek jelennek meg

FLOYD-WARSHALL ALGORITMUS D ÉS π MÁTRIXOK

$$D_{ij}^{(0)} = \begin{cases} 0, \text{ ha } i = j \\ w(i,j), \text{ ha } (i,j) \in G.E \land i \neq j \\ \infty, \text{ ha } i \neq j \land (i,j) \notin G.E \end{cases}$$

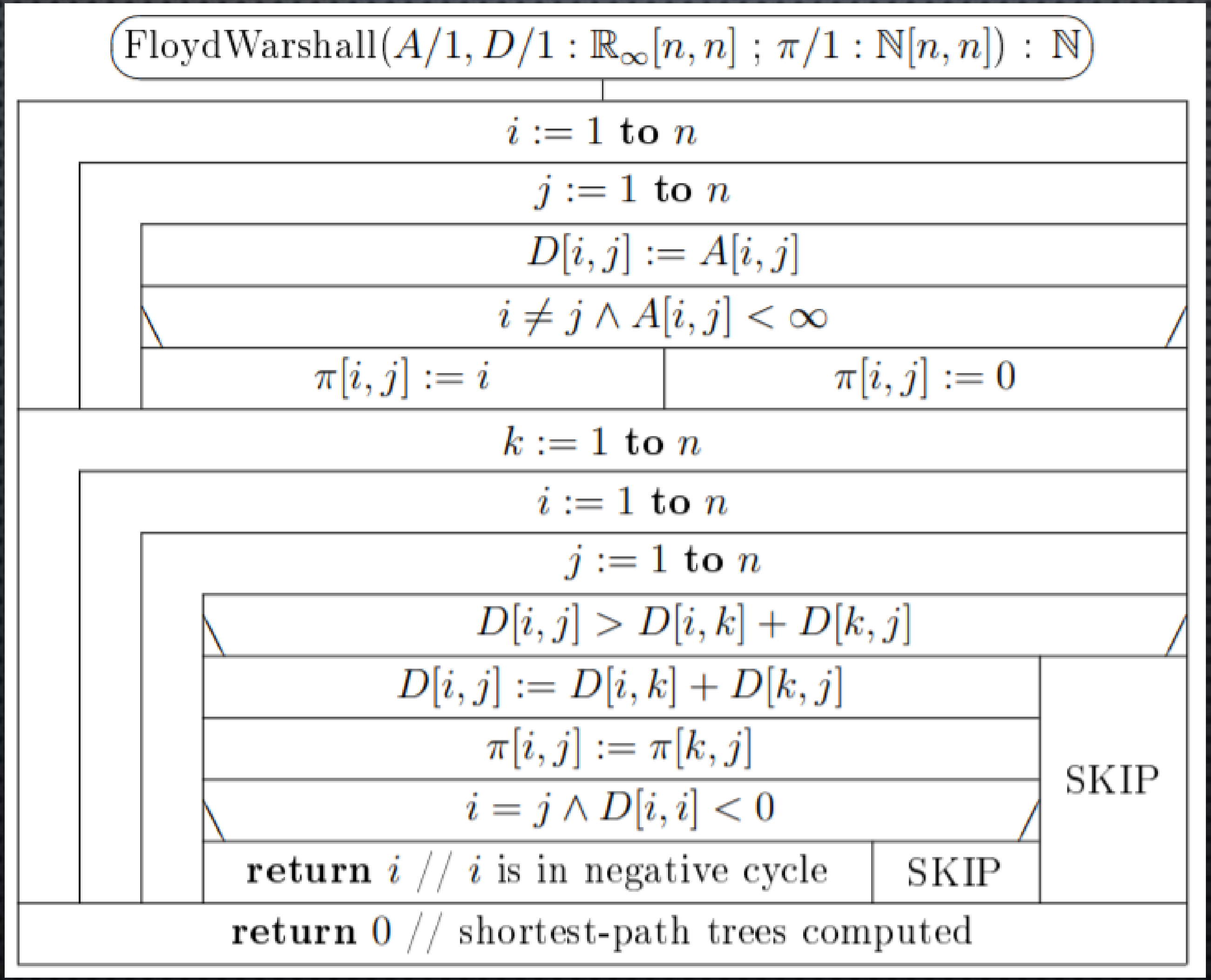
$$\pi_{ij}^{(0)} = \begin{cases} 0, \text{ ha } i = j \\ i, \text{ ha } (i,j) \in G.E \land i \neq j \\ 0, \text{ ha } i \neq j \land (i,j) \notin G.E \end{cases}$$

$$D_{ij}^{(k)} = \begin{cases} D_{ik}^{(k-1)} + D_{kj}^{(k-1)}, \text{ ha } D_{ij}^{(k-1)} > D_{ik}^{(k-1)} + D_{kj}^{(k-1)} \\ D_{ij}^{(k-1)} \text{ különben} \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{kj}^{(k-1)}, \text{ ha } D_{ij}^{(k-1)} > D_{ik}^{(k-1)} + D_{kj}^{(k-1)} \\ \pi_{ij}^{(k-1)} \text{ különben} \end{cases}$$

Elég egy mátrixpár, a k. sor és oszlop nem változik a (k-1). és k. lépés között

FLOYD-WARSHALL ALGORITMUS



$\Theta(n^3)$

ISMÉTLÉS: CYK ALGORITMUS

- CYK parser
- Adott egy Chomsky 2-es típusú CNF grammatika és egy input szó
- Kérdés, hogy a szó levezethető-e a grammatika szabályaival
- Szintaxis fa építés

	1	2	3	4	5	6
6	S					
5	S	S				
4	B	∅	B			
3	∅	∅	∅	∅		
2	A, U	∅	V	∅	C, W	
1	A, X	A, X	Z	Z	C, Y	C, Y
	a	a	b	b	c	c