

beadási:

megoldás: út grafikon

Input: G elszigetelt gráf
 s kezdőcsomópont

Feladat: s -ből v -re
 megoldás: út és ennek
 hossza

legáltalánosabb esetben (Bellman-Ford) is kell egy
 megközelítés: ne legyen negatív kör!

J-F: elve: $T[i, v] = \min_{u \in V} \{T[i, u] + c(u, v)\}$ ahol $c(u, v)$ az u -ból v -re
 az i -edik lépésben $i = 1, 2, \dots, n-1$

$$T[1, v] = \begin{cases} 0 & \text{ha } v = s \\ c(s, v) & \text{ha van } s \rightarrow v \text{ éle} \\ \infty & \text{egyébként} \end{cases}$$

$$P[1, v] = s \quad \forall v \in V$$

$$P[i, v] = \min_{u \in V} \{P[i, u] + c(u, v)\}$$

ha nincs negatív
 kör akkor nem
 lehet kör ami
 rövidítaná az
 utat...

$n-1$ -ig kell
 megismételni

utak
 lehet hossz
 több csomópont
 feladat en,

elszigetelt
 egyenlő algoritmus
 futtatása és
 algoritmus

utak
 Neptunban

előző nap deludó
 munka konzultáció

megr. ① ha $T[i+1, v] = T[i, v]$ akkor nem kell folytatni az algoritmust, innentől nincs rövidítés

② Ha az utolsó $n-1$ sor után:

1.) $T[n, v] = T[n-1, v]$ ~~($\forall v \in V$)~~ $\forall v \in V$

$\Rightarrow$ biztos nincs negatív kör

2.) $T[n, v] \neq T[n-1, v]$ van kisebb érték

$\Rightarrow$ legfeljebb n élű út rövidített $\Rightarrow$

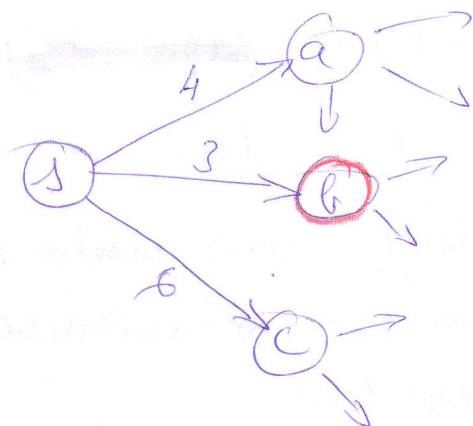
$\Rightarrow$ van negatív kör

Legrövidebb út keresése gráfban:

Név	Mikor	LSz
BFS	nincs súly (minden ≥ 1)	$O(n^2) / O(n \cdot e)$
legrövidebb út keresés DAG-ban	ha a gráf DAG (nincs benne kör)	$O(n^2) / O(n \cdot e)$
Bellman-Ford	ha nincs negatív kör	$O(n^3) / O(n \cdot e)$
Dijkstra	minden élsúly nem negatív ("elég gyors" problema)	$O(n^2) / O(e \cdot \log(n))$

I.e. van olyan hegy negatív a körben de pl.: tördelés
negatív kör már működik

Tfh.: $\exists d(t) \geq 0 \quad \forall t \in V$



Δ -ből legközelebbi út:

híttas, mert b -be
nincs tovább út 3-nál
mert ha nincs negatív
él, akkor a -n
és c -n keresztül
bármely út b -be
egyalább 4 és 6

1.) Állítás

s és v

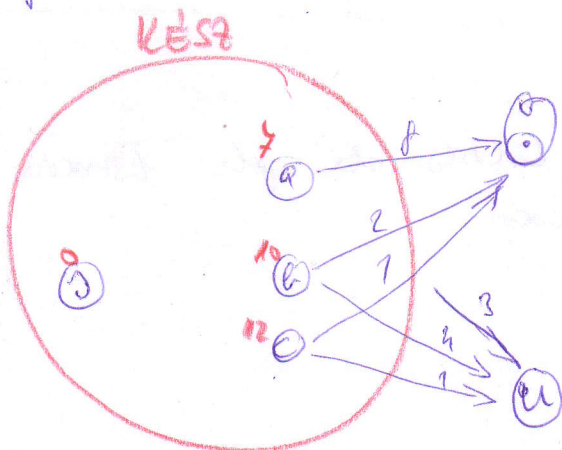
~~átmeneti állítás~~

~~KÉSZ~~ : az az ennélféle
amire ismert a
minimális út

$$2.) \text{tdvolsag}[v] = \begin{cases} \text{min út } v\text{-be ha } v \in \text{KÉSZ} \\ * & \text{ha } v \notin \text{KÉSZ} \end{cases}$$

3.) $d[v] =$ legközelebbi út s -ből v -be
ami végig ~~KÉSZ~~ KÉSZ-ben halad
érvényes az utolsó él

pl:



$$d[v] = 12$$

$$d[u] = 10$$

ha csak u és v
van olyan ami
előrehalad KÉSZ-ből
akkor u tdvolsaga
létezik ami ismert

Algoritmus:

$$KESZ = \{s\}$$

$$tdolsag[v] = \begin{cases} 0 & \text{ha } v=s \\ * & \text{egyébent} \end{cases}$$

$$d[v] = \begin{cases} * & \text{ha } v=s \\ c(s,v) & \text{ha van } s \rightarrow v \text{ él} \\ \infty & \text{egyébent} \end{cases}$$

honnau[v] = s $\forall v \in V$ # honnan jött az eddig ismert legrövidebb út s-ből v-he

while van olyan v csúcs amire $d[v] \neq *$ és $d[v] \neq \infty$
 v^* az a csúcs amire $d[v]$ minimális:

$$v^* \rightarrow KESZ$$

$$tdolsag[v^*] = d[v^*]$$

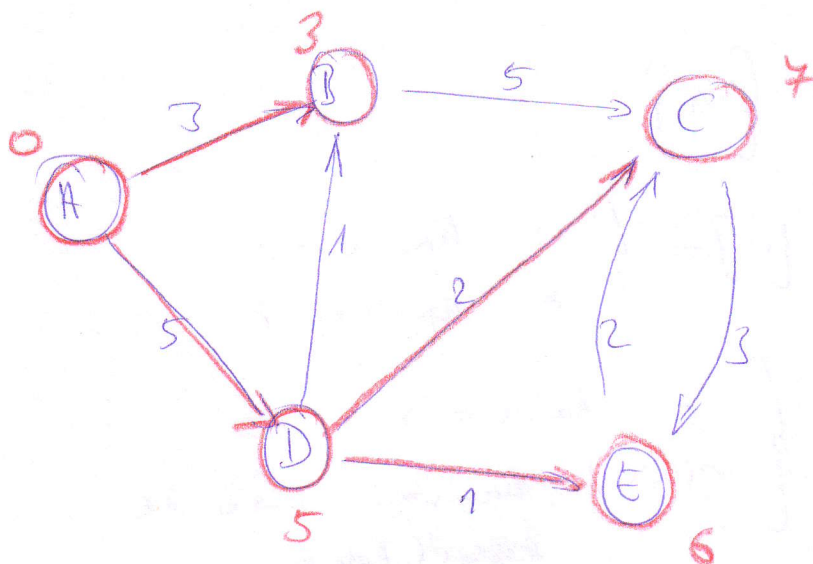
$$d[v^*] = *$$

$d[w]$ egyenértékűen minden w -re ami nincs benne a ~~KESZ-ben~~ $KESZ$ -ben és v^* -nak közvetlen
akár, ha $tdolsag[v^*] + c(v^*, w) < d[w]$

$$\text{akkor } d[w] = tdolsag[v^*] + c(v^*, w)$$

$$\text{honnau}[w] = v^*$$

pelda:



Edvolyék

A	B	C	D	E
0	*	*	*	*
0	3	*	*	*
0	3	*	5	*
0	3	*	5	6
0	3	7	5	6

Késsz

d

A	B	C	D	E
*	3	∞	5	∞
*	*	8	5	∞
*	*	7	*	6

0 = A

A, B

A, B, D

A, B, D, E

honnán:

A	B	C	D	E
A	A	A	A	A
A	A	A	A	A
A	A	D	A	D
A	A	D	A	D