

Algoritmusok és adatszerkezetek 2.

Fekete István előadása alapján

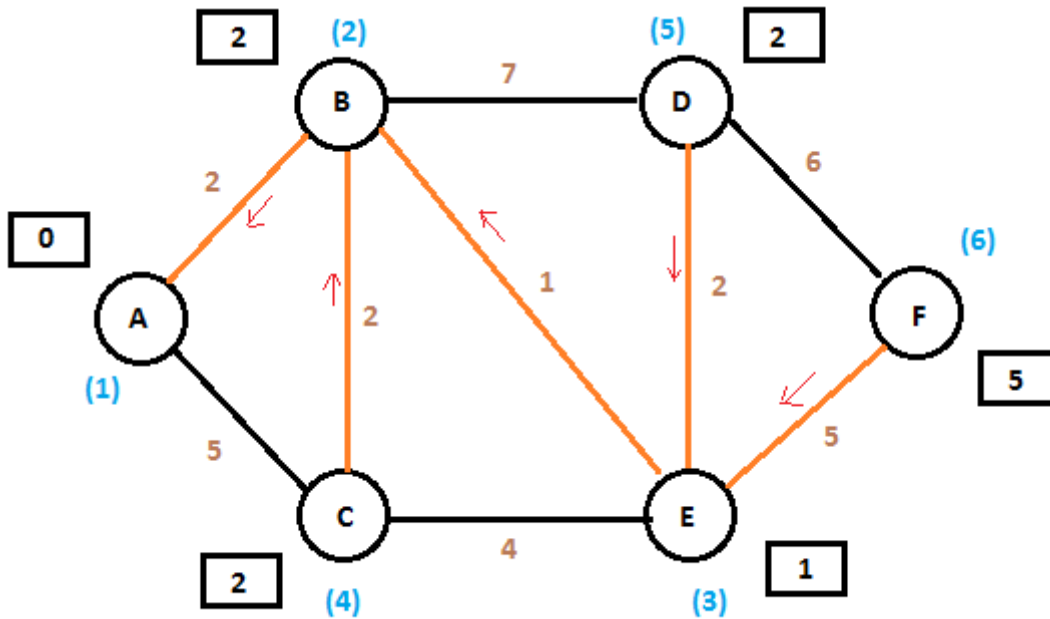
Készítette: Nagy Krisztián

6. előadás

26. Minimális költségű feszítőfák, Prim-algoritmus

(A piros-lél algoritmus nem tananyag)

Például: Elektromos áramot szeretnénk bevezetni



$G = (V, E)$ irányítás nélküli gráf

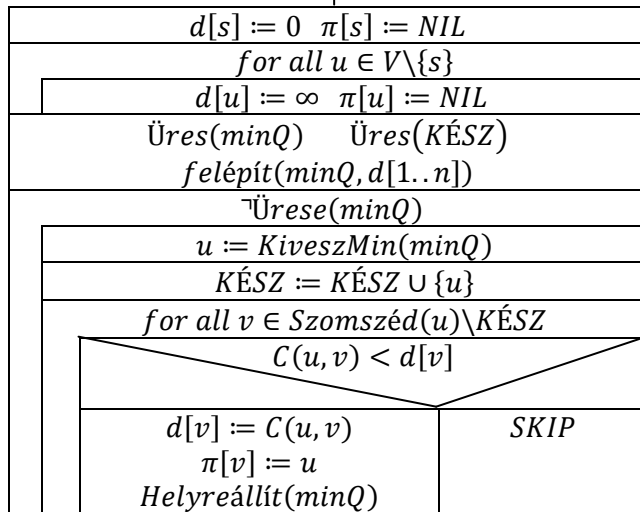
s start csúcs szabadon választható

Feladat: Olyan feszítőfa meghatározása, amely összélköltsége minimális.

Megjegyzés: Más a költség függvény, mint a Dijkstra-nál.

Dijkstra-nál: $d[u] + C(u, v)$, itt $C(u, v)$

$Prim(G, s)$ -s szabad választás eredménye



Művelet igények implementációk alapján:

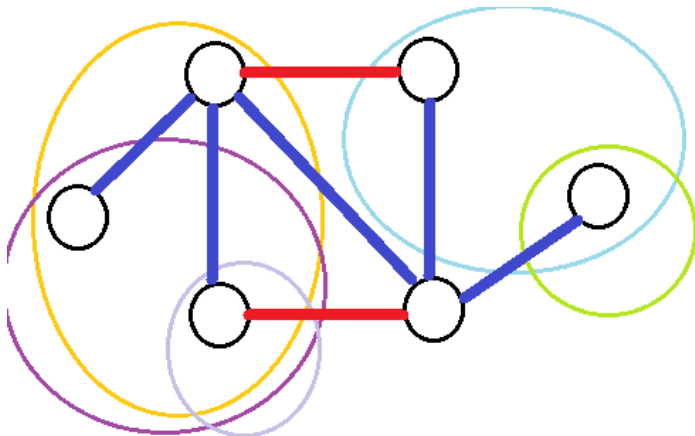
1. implementáció: $minQ, d[1..n]$: Feltételes minimum keresés
 $T(n) = \theta(n^2 + e) = \theta(n^2)$
2. implementáció: $minQ$: kupac
 $T(n) = \theta(n \log n) + O(e \log n) = \theta((n + e) \log n)$
 Sűrű gráfok esetén, ezt az implementációt ne használjuk!

Piros- kék algoritmus (Nem tananyag)

Három híres minimum költségű feszítőfát kereső algoritmus általános algoritmus adja.

Kék szabály: $X \subseteq V$, amelyből nem megy ki kék él: $min\{d(e)\} \rightarrow$ kék lesz

Piros szabály: Egyszerű kör választása, melyben nincs piros él: $max\{d(e)\} \rightarrow$ piros lesz

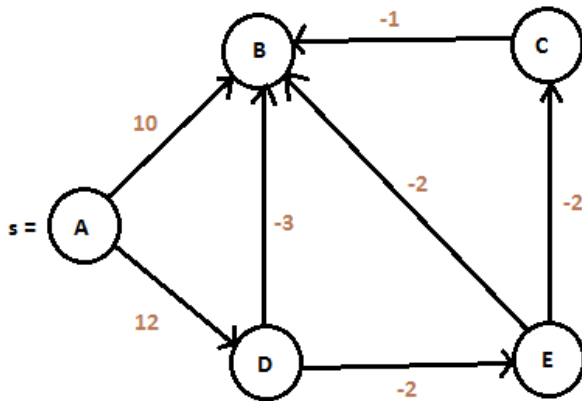


Prim: Csak kék szabály. $X :=$ összes nyílt csúcs

25. Legrövidebb utak egy forrásból II. Bellman-Ford algoritmus (negatív élköltség)

Negatív élköltség megengedett, viszont a negatív kört kizárjuk!

Például:



Nem mohó algoritmus! A Dijkstra nem jó!

G gráf irányított

Milyen hosszú lehet a minimális költségű $s \rightsquigarrow u$ út, adott u csúcs esetén?

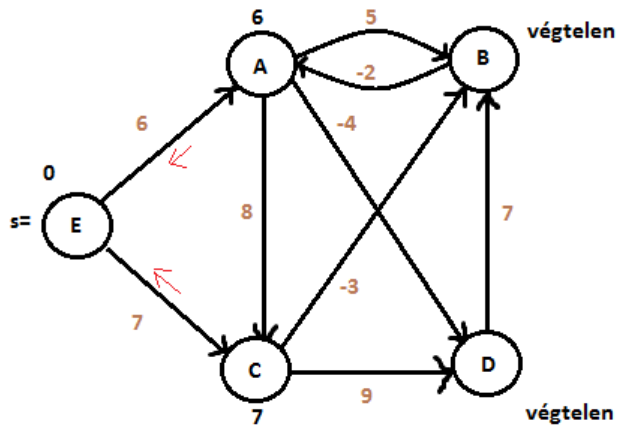
Válasz: Maximum $n - 1$ hosszú út lehet (Mivel nem tartalmaz kört!) \Rightarrow Próbáljunk meg $(n - 1)$ -szer közelíteni a teljes gráfon

Bellman – Ford(G, s)

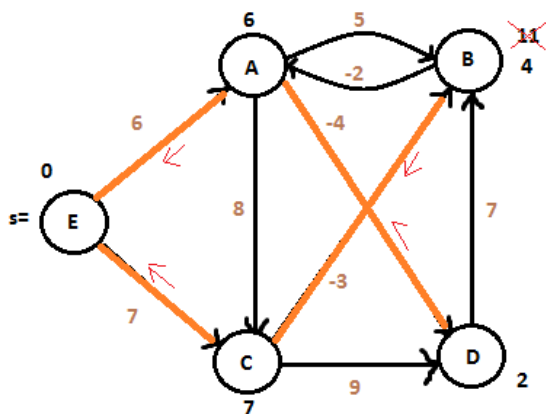
$d[s] := 0 \quad \pi[s] := NIL$	
for all $u \in V \setminus \{s\}$	
$d[u] := \infty \quad \pi[u] := NIL$	
for $i = 1$ to $n - 1$	
for all $(u, v) \in E$ (élre)	
$d[u] + C(u, v) < d[v]$	
$d[v] := d[u] + C(u, v)$	SKIP
$\pi[v] := u$	

Példa + 1. iteráció:

Megállapodás: Élsorrend: AB,AC,AD,BA,CB,CD,DB,EA,EC



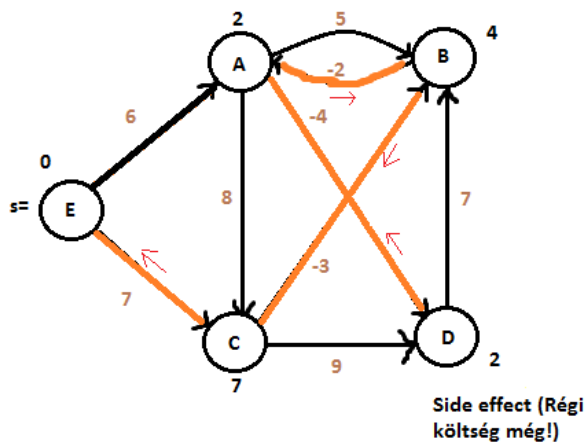
2. iteráció:



Invariáns a k -adik menetben megtaláljuk minden csúcshoz a legkisebb költségű legfeljebb k élet tartalmazó utat.

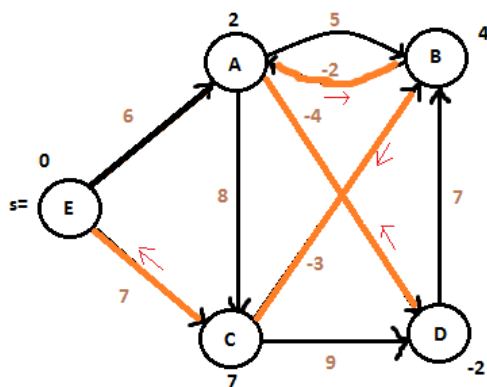
Próba: EADB $9 \nless 4 \rightarrow$ Nem javít!

3.iteráció:



Legjobb 3 hosszú utak (+ side effect ☺)

4.iteráció:



Negatív kör kizárása? Az irodalomban be van építve az algoritmusba az n-edik menet a ciklusban (Ugye eredetileg n-1-ig megyünk algoritmusunkban). Amennyiben az n-edik menetben változás történt, akkor baj van. (Negatív kör)