Gráfalgoritmusok

Gaskó Noémi

2023. március 5.

Tartalomjegyzék

Múlt órán

- követelmények
- alapfogalmak
- gráfok ábrázolása

Milyen feladatok merülnek fel gráfokkal kapcsolatban?

- gráfok bejárása

Milyen feladatok merülnek fel gráfokkal kapcsolatban?

- gráfok bejárása
- összefüggő-e

Milyen feladatok merülnek fel gráfokkal kapcsolatban?

- gráfok bejárása
- összefüggő-e
- utak keresése

Milyen feladatok merülnek fel gráfokkal kapcsolatban?

- gráfok bejárása
- összefüggő-e
- utak keresése
- ...

Gráfok bejárása

- mélységi bejárás (DFS - depth-first search)- mint egy séta

Gráfok bejárása

- mélységi bejárás (DFS - depth-first search)- mint egy séta
- szélességi bejárás (BFS - breadth-first search)- mint ahogy a hullám terjed

Szélességi bejárás (Breadth-first search BFS)

- több algoritmus a BFS ötletén alapul (pl. Dijkstra, Prim)

BFS algoritmus

Adott egy $G = (V, E)$ gráf és egy **kiinduló csomópont** s, az algoritmus bejárja a gráfot, hogy minden csomópontot felfedezzen, ami elérhető az s-ből

- irányított és nem irányított gráfok esetén is működik
- az algoritmus egy fát épít fel, melynek gyökere az s

BFS (II)

- ahhoz, hogy megfigyeljük hogyan működik az algoritmus három szinre van szükségünk: fehér, szürke és fekete:
  - fehér - nem volt meglátogatva
  - fekete - ha $\{u, v\} \in E$, u fekete, akkor v fekete vagy szürke
  - szürke - a meglátogatott és nem meglátogatott csomópontok "határán" van

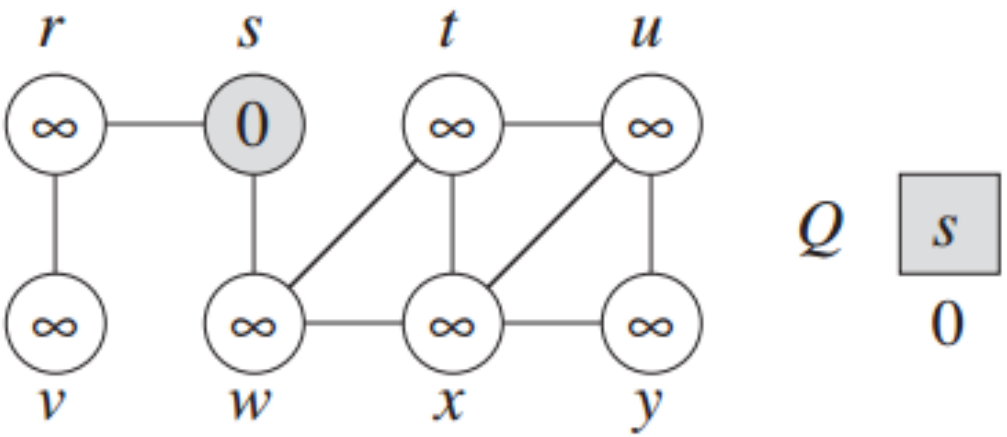
BFS (III)

- az eljárás adiacencia listával dolgozik
- a π érték a szülőt, a d érték a távolságot tartja nyilván a kezdeti csomópontból az aktuális csomópontba

BFS(G,s)

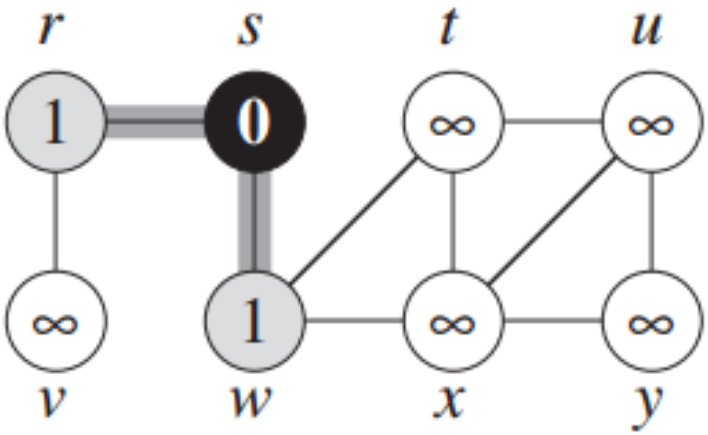
For minden csomópont $u \in G, V - \{s\}$
    u.color = feher
    u.d = ∞
    u.π = NIL
s.color = szurke
s.d = 0
s.π = NIL
$Q = \varnothing$
Enqueue(Q,s)
While $Q \neq \varnothing$
    u = Dequeue(Q)
    For minden $v \in G.Adj[u]$
        If v.color == feher
            v.color = szurke
            v.d = u.d + 1
            v.π = u
            Enqueue(Q,v)
    u.color = fekete

Egy példa[1]



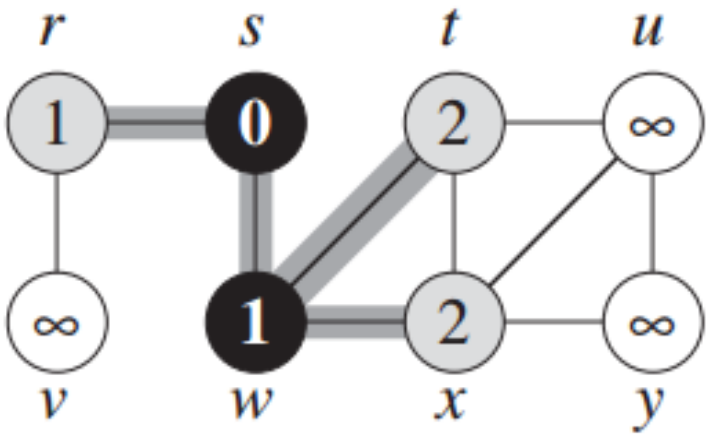
[1] forrás: Cormen, Introduction to algorithms

Egy példa (folyt.)



Q | w | r |
|---|---|
| | 1 | 1 |

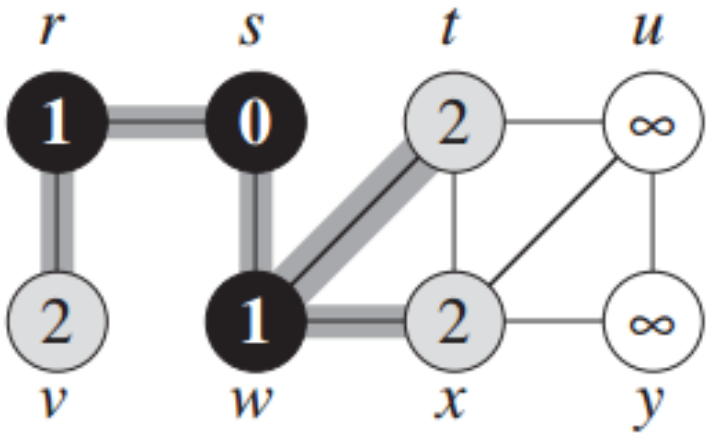
Egy példa (folyt.)



Q

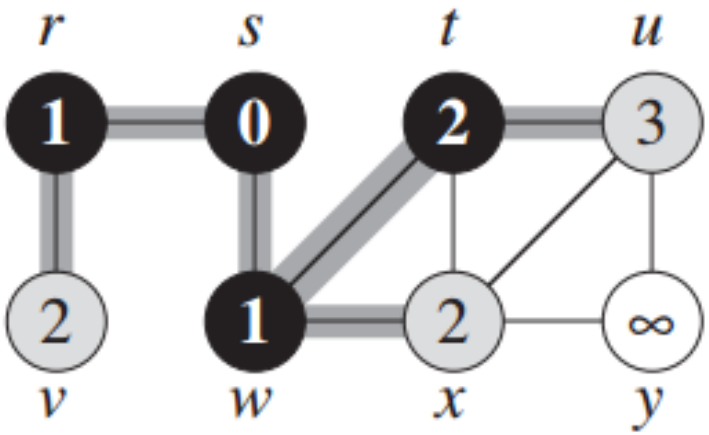
r	t	x
1	2	2

Egy példa (folyt.)



Q | t | x | v |
|---|---|---|
| 2 | 2 | 2 |

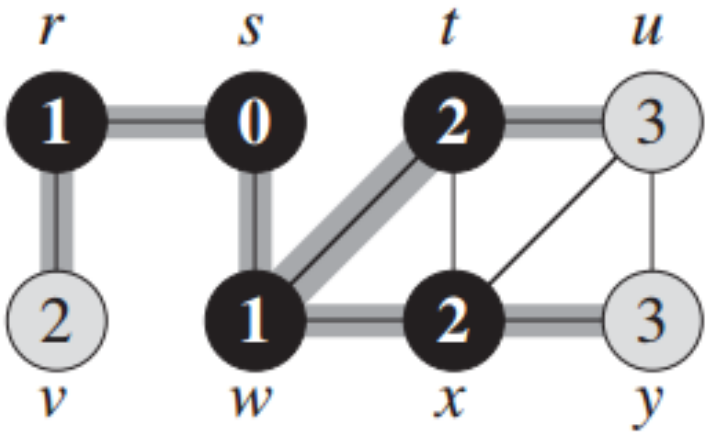
Egy példa (folyt.)



Q

x	v	u
2	2	3

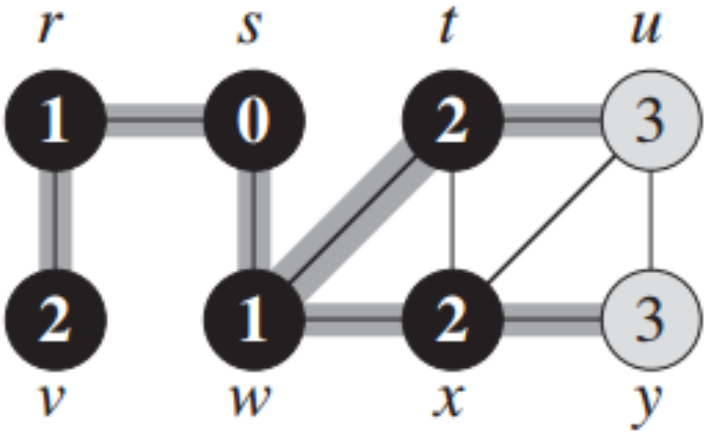
Egy példa (folyt.)



Q

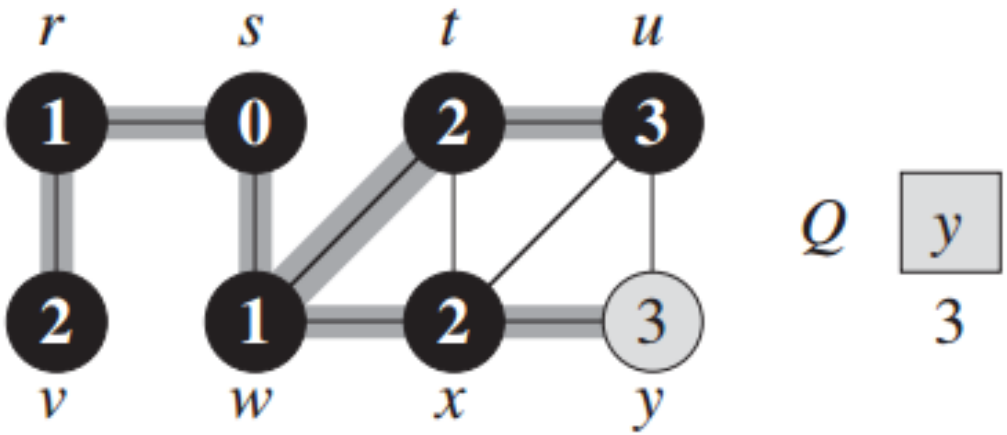
v	u	y
2	3	3

Egy példa (folyt.)

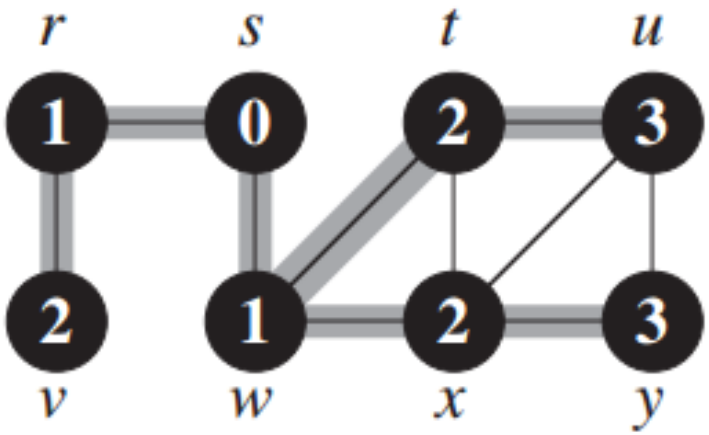


Q | u | y |
|---|---|
| 3 | 3 |

Egy példa (folyt.)



Egy példa (folyt.)



$Q \quad \emptyset$

Szélességi bejárás - nem jelölve a csomópontokat

```
BFS(G,source):
(source a kezdeti csomópont)
create a queue Q (várakozási sor)
enqueue source onto Q
mark source
while Q is not empty:
        dequeue an item from Q into v
        for each edge e incident on v in Graph:
            let w be the other end of e
            if w is not marked:
                mark w
                enqueue w onto Q
```

- az algoritmus végrehajtási ideje $O(V + E)$

Mélységi bejárás (Depth-first search DFS)

- miután minden élt bejárt, amit a v-ből elért, visszatér annak az élnek a csomópontjára, amely a v -be ment, és onnan folytatja a bejárást

DFS (II)

- itt is kiszinezzük a csomópontokat, mint a BFS esetén
- megjegyezzük azt is, hogy mennyi időbe telt felfedezni
- $u.d$ az időt jelzi, amikor felfedeztük u-t
- $u.f$ amikor "befejeztük" u
- a csomópont állapota: fehér - $u.d$ - szürke - $u.f$ - fekete

DFS - az eljárás

DFS(G, s)
For minden csomópont $u \in G.V$
    u.color = feher
    u.d = ∞
    u.π = NIL
time = 0
For minden $u \in G.V$
    If u.color == feher
        DFS_VISIT(G,u)

DFS - (II)

DFS_VISIT(G, u)
$time = time + 1$
$u.d = time$
u.color = szurke
For minden $v \in G.Adj[u]$
  If v.color == feher
    $v.\pi = u$
    DFS_VISIT(G,v)
  u.color = fekete
  $time = time + 1$
  $u.f = time$

DFS

- az algoritmus végrehajtási ideje:
  - DFS_VISIT végrehajtódik $|Adj[v]|$-szor, mert

$$\sum_{v \in V} |Adj[v]| = \Theta(E)$$

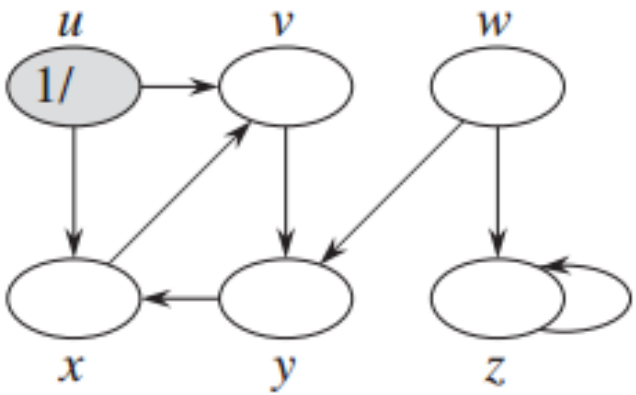
  - az algoritmus végrehajtási ideje $\Theta(V + E)$

Élek típusai

- fa él (Tree edge) - (u, v) egy fa él, ha u után v-t látogatjuk meg
- vissza él (back edge) - (v, u) egy él, de nem tartozik hozzá a DFS fához
- előre él (forward edge) - v-t az u után látogatjuk meg, de az (u, v) él nincs benne a DFS fában
- kereszt él (cross edge) - az egyéb élek

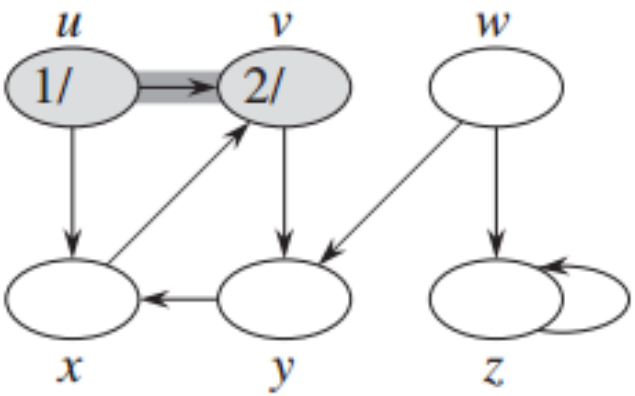
Példa: lásd 2_jegyzet.pdf

Egy példa[2]

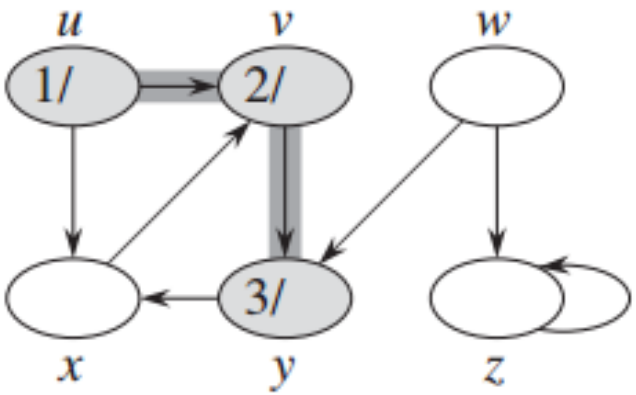


[2] Cormen, Introduction to Algorithms

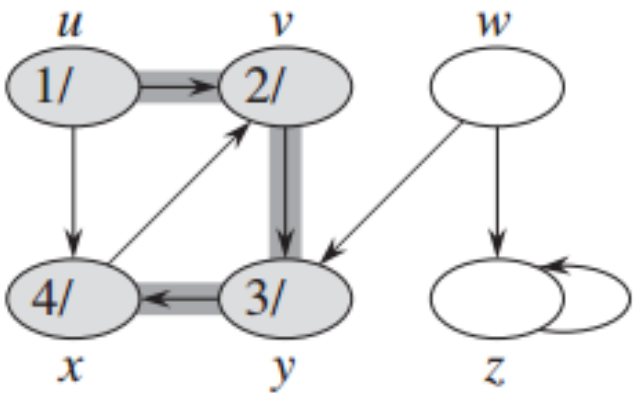
Egy példa (folyt.)



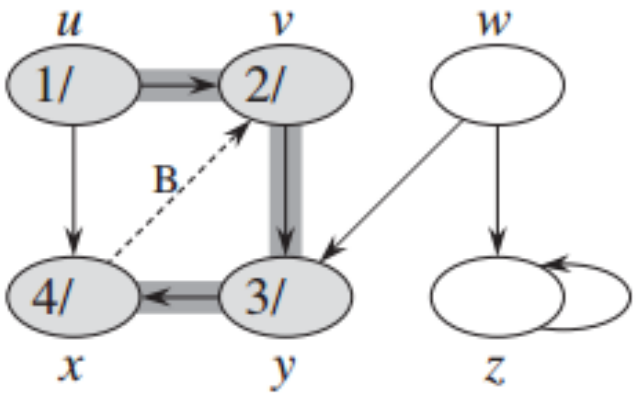
Egy példa (folyt.)



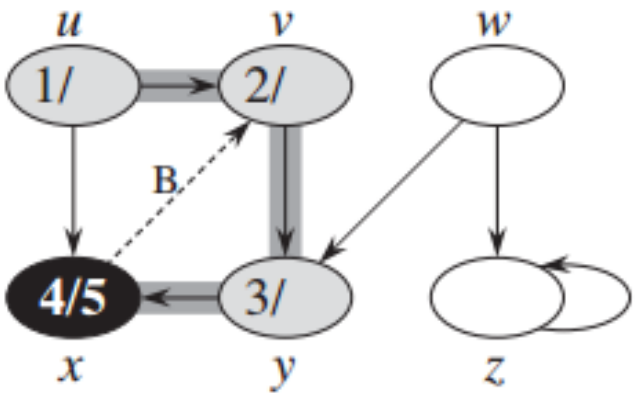
Egy példa (folyt.)



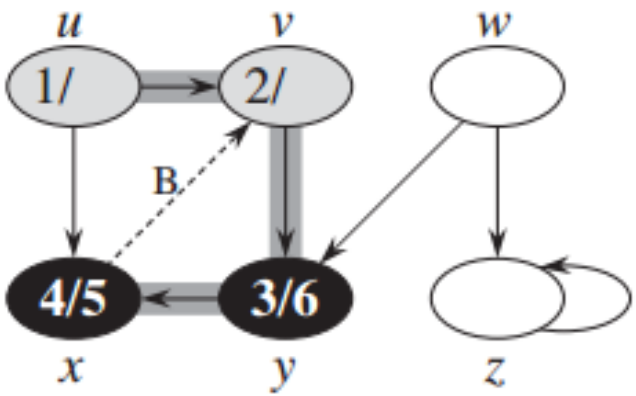
Egy példa (folyt.)



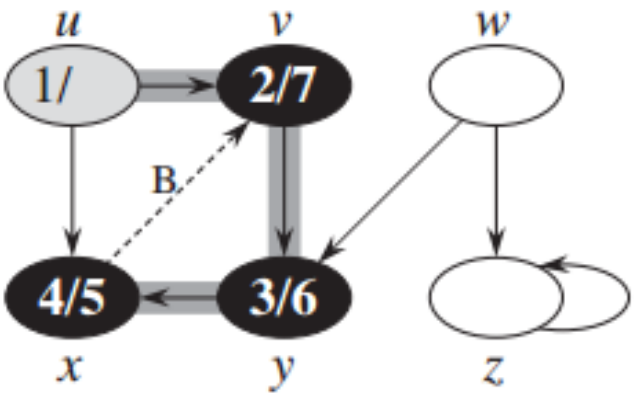
Egy példa (folyt.)



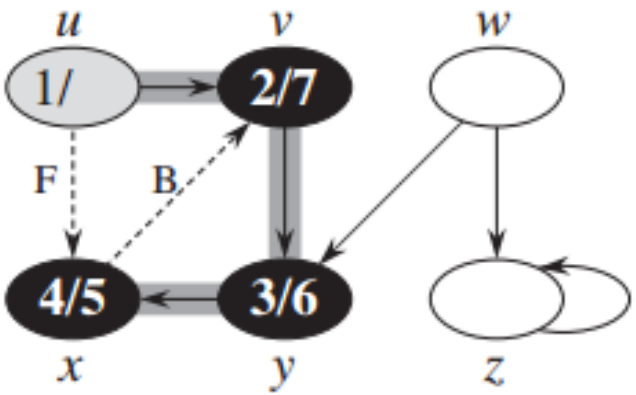
Egy példa (folyt.)



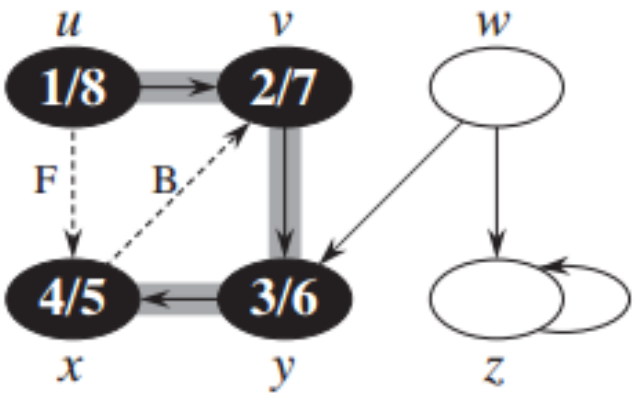
Egy példa (folyt.)



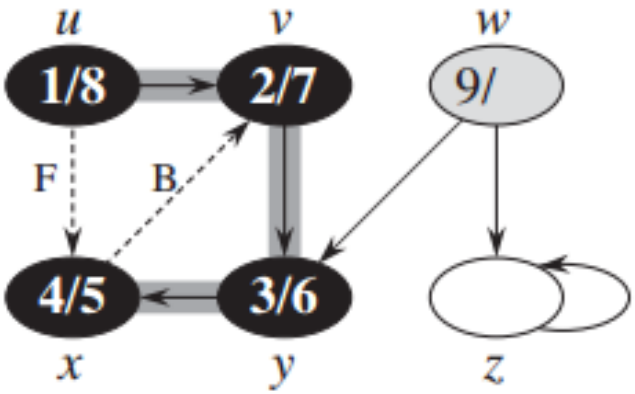
Egy példa (folyt.)



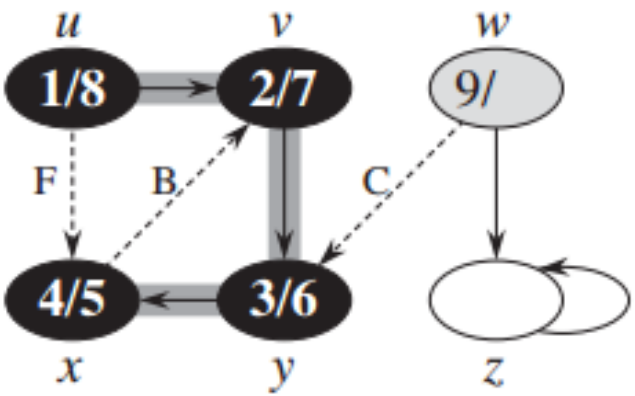
Egy példa (folyt.)



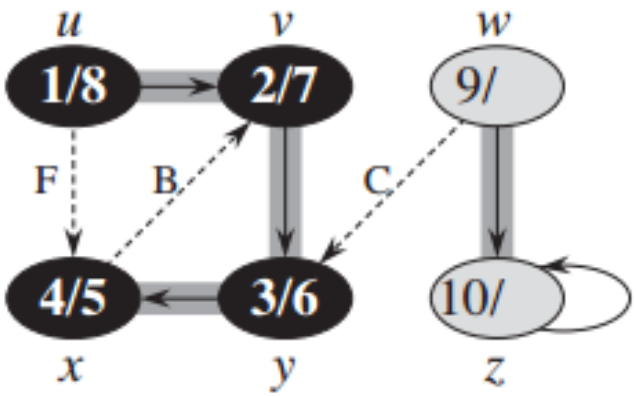
Egy példa (folyt.)



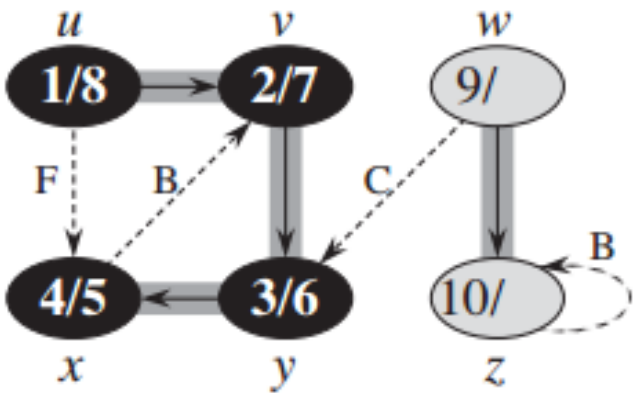
Egy példa (folyt.)



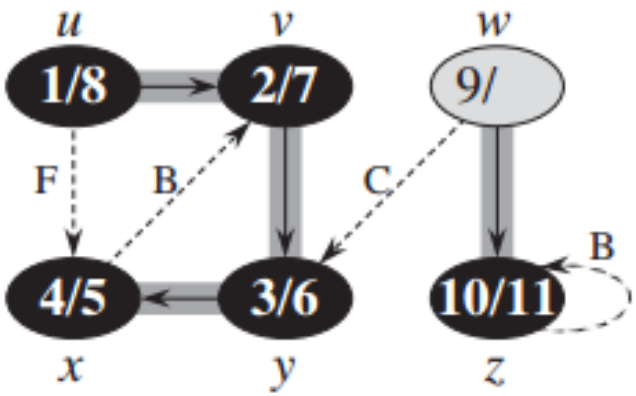
Egy példa (folyt.)



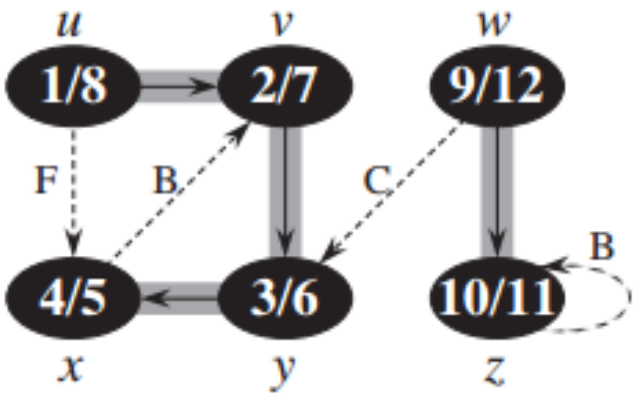
Egy példa (folyt.)



Egy példa (folyt.)



Egy példa (folyt.)



Bejárások - irányított gráfok

Bejárások: DFS és BFS hasonlóan, mint a nem irányított esetben

Bejárások - irányított gráfok

Bejárások: DFS és BFS hasonlóan, mint a nem irányított esetben

Összefüggő komponens meghatározása

Összefüggőség irányítatlan gráfban

Egy irányítatlan gráfot összefüggőnek nevezünk, ha bármely két csúcsa között létezik út.

Összefüggő komponens meghatározása

Összefüggőség irányítatlan gráfban

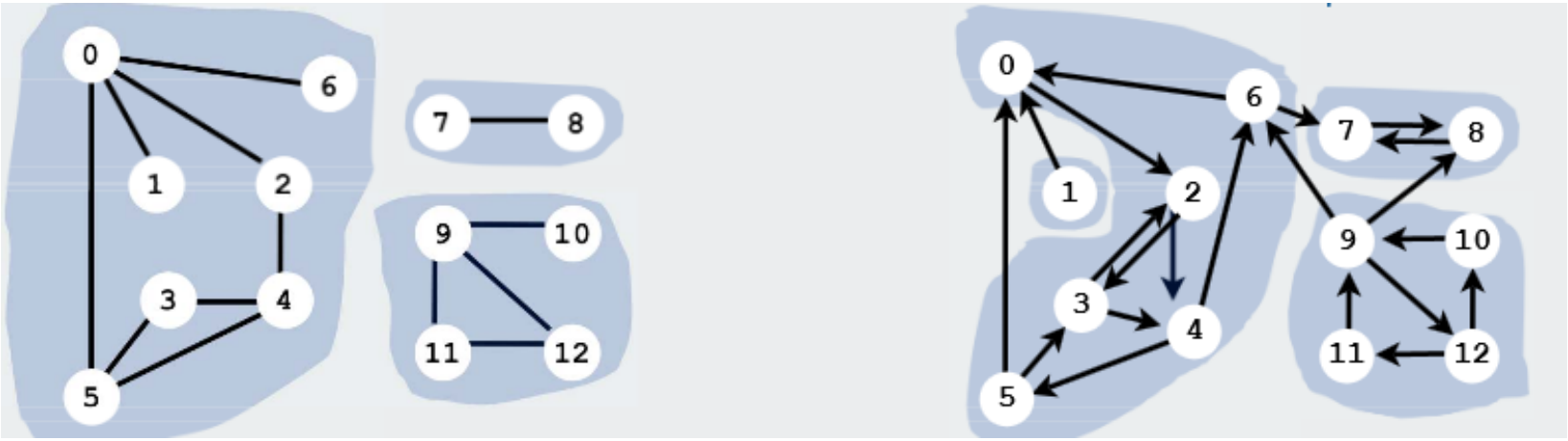
Egy irányítatlan gráfot összefüggőnek nevezünk, ha bármely két csúcsa között létezik út.

Összefüggőség irányított gráfban

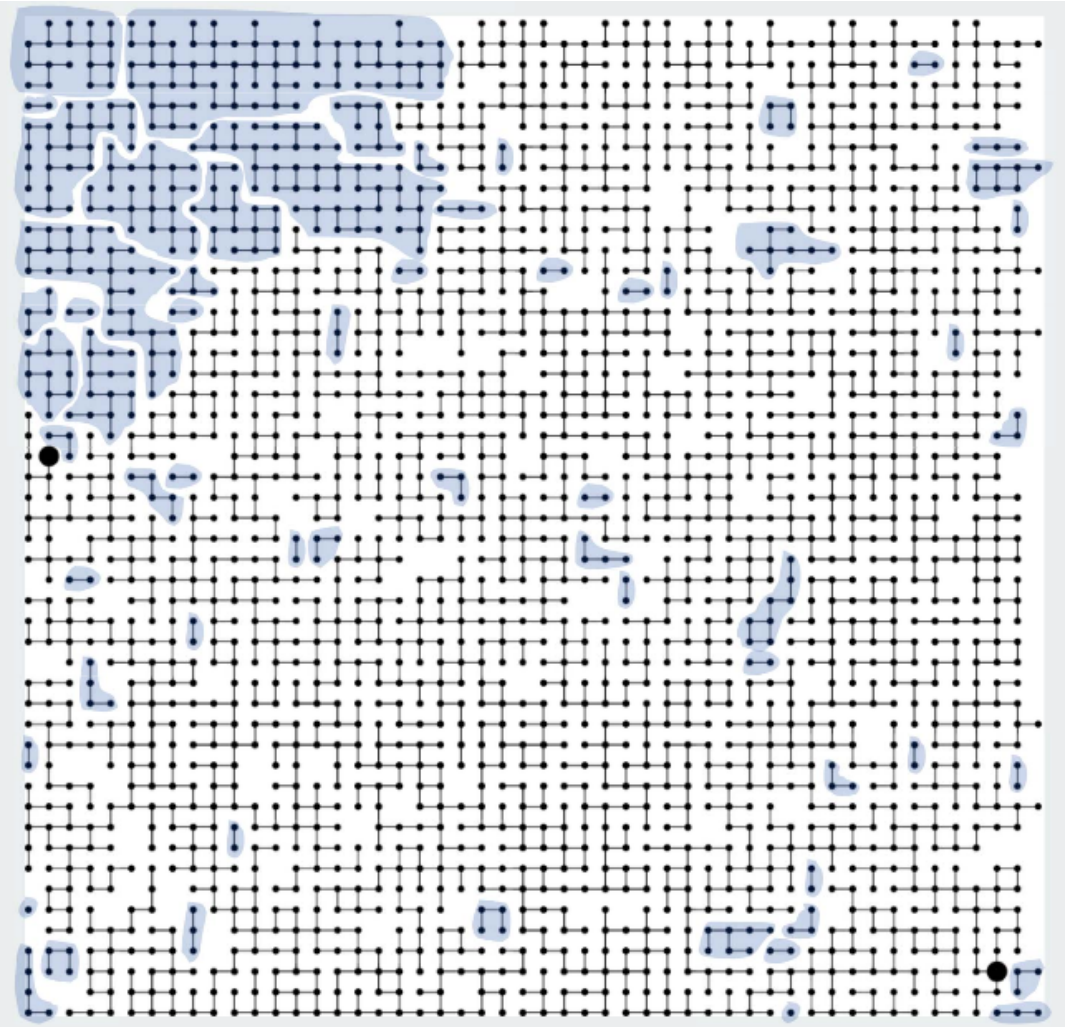
Egy irányított gráfot gyengén összefüggőnek nevezünk, ha a neki megfelelő irányítatlan gráf összefüggő.
Egy irányított gráf erősen összefüggő ha bármely u és v csúcsra létezik $u \to v$ irányított út.

Példa

Összefüggõség:



Hány összefüggő komponense van az alábbi ábrának :)?



Erősen összefüggő komponensek meghatározása - "+-"-algoritmus

Gráf transzponáltja

Legyen G^T a G gráf transzponáltja, melyet úgy kapunk, hogy minden él irányítását megfordítjuk.

Az algoritmus alapötlete:

- inditsunk egy mélységi bejárást v-ből, és minden csúcspont ahova eljutottunk jelöljük meg "+"-al
- transzponáljuk a gráfot
- indítsunk még egy bejárást v-ből a transzponált gráfon, jelöljük "-"-al ahova eljutottunk
- v erősen összefüggő komponensei azok amik + és - jelet is tartalmaznak
- ismételjük az eljárást a maradék csomópontokra

Erősen összefüggő komponensek meghatározása - Kosaraju algoritmusa

Kosaraju algoritmusa
Az algoritmus lépései:

- bejárjuk a gráfot DFS-el - a bejárt elemeket egy verembe tesszük
- bejárjuk a gráf transzponáltját DFS-el a verem alapján

Egy példa - lásd 2_jegyzet.pdf

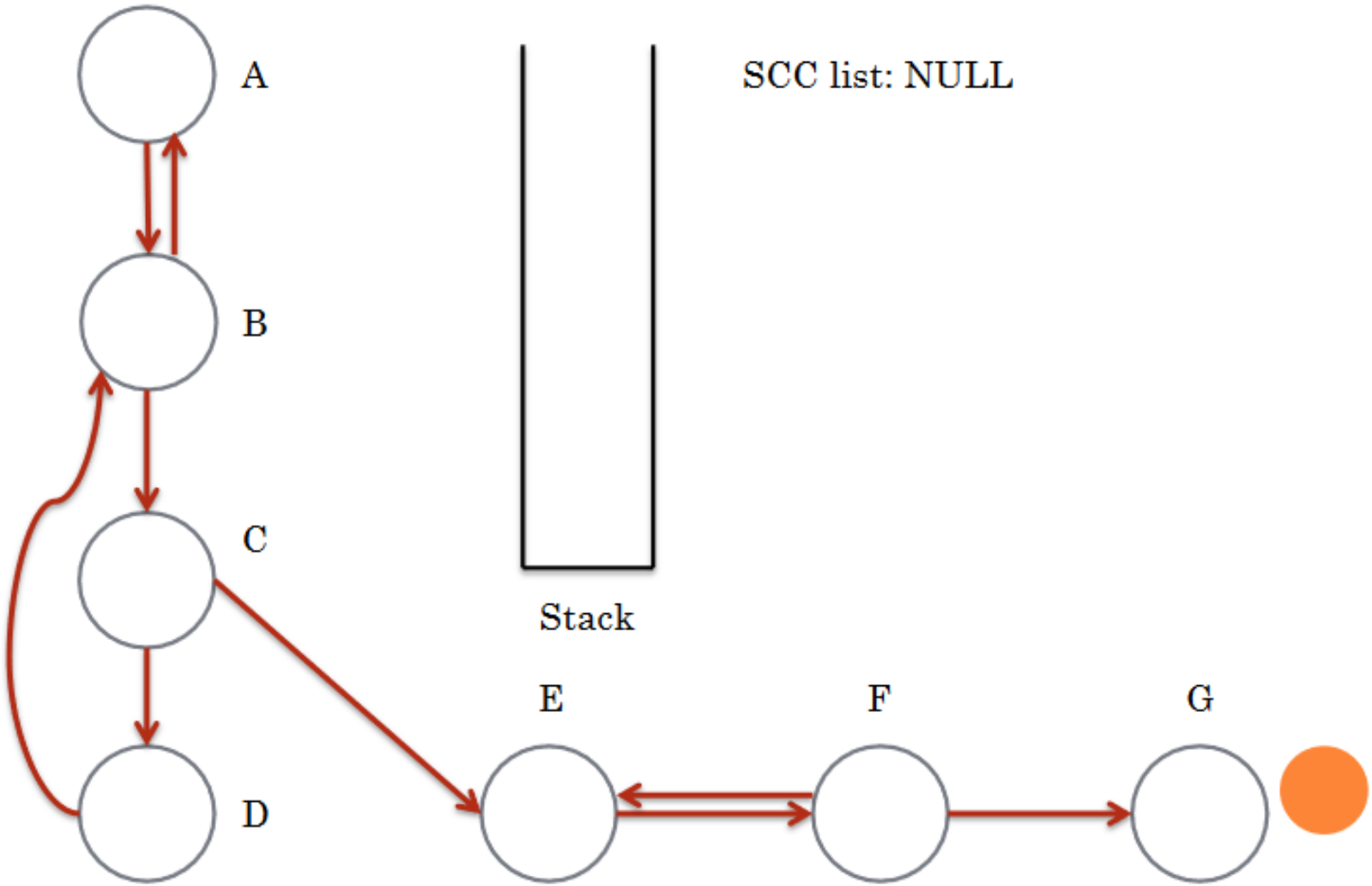
Erősen összefüggő komponensek meghatározása - Tarjan algoritmusa

- csak egy mélységi bejárást használ
- minden csomópont esetén tároljuk a következőket:
  1. az időpillanatot, amikor meglátogattuk (amikor szürke lett) (time)
  2. v a segéd vermen van-e
  3. a v-ből elérhető legkorábban meglátogatott csúcsok belépési ideje (low[v])

DFS_SCC(v)

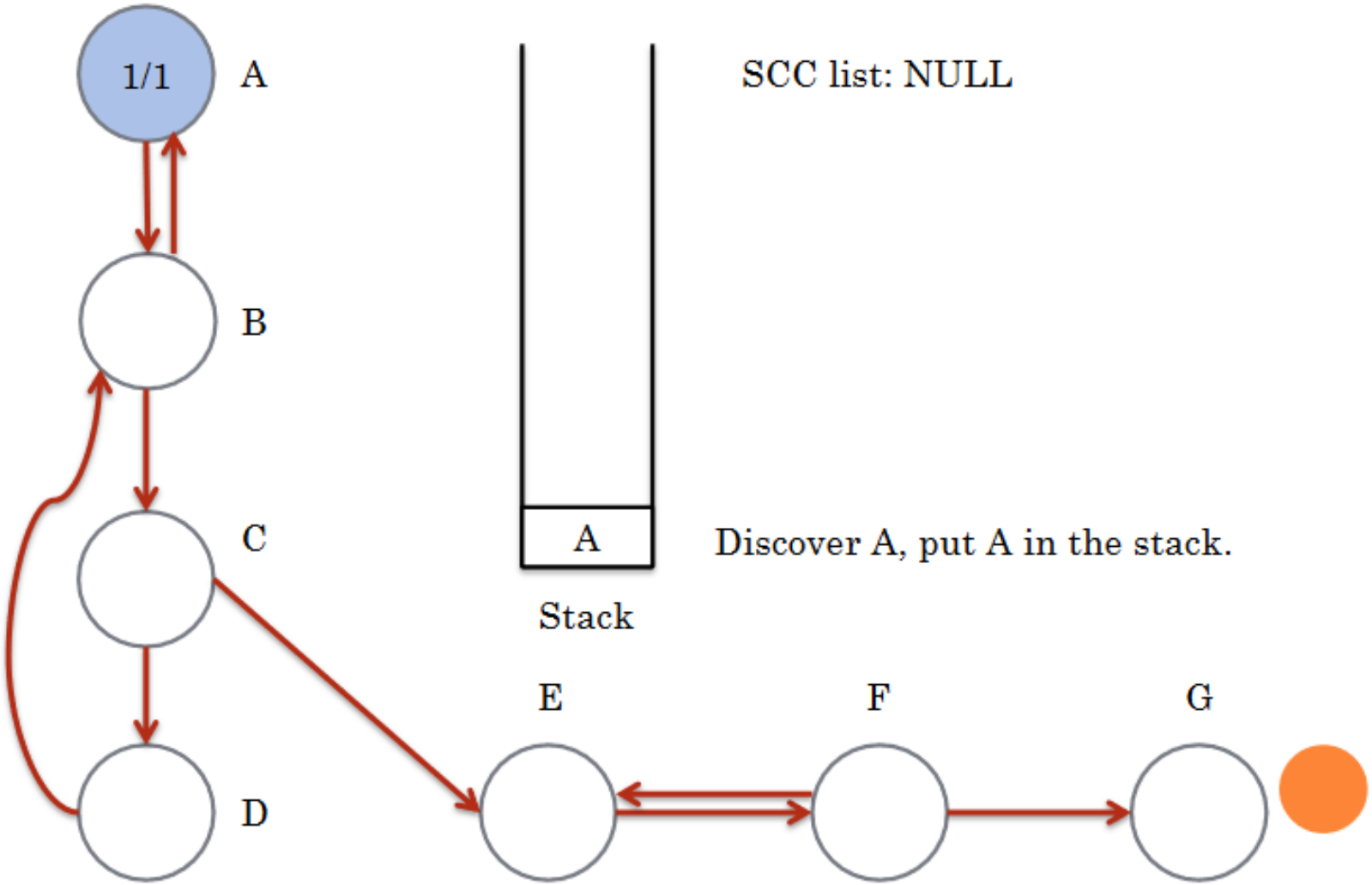
```
time ++
belep[v]=time
low[v]=time
vermen[v]= TRUE
FOR minden w szomszédjára v-nek
   IF (belep[w]=-1)
      DFS_SCC(w)
      low[v]=min(low[v],low[w])
   ELSE
      IF ((belep[w]<belep[v]) AND vermen[w])
         low[v]=min(low[v],belep[w])
IF (low[v]=belep[v])
   komponensek++
   WHILE ((verem.empty()=FALSE) AND (belep[verem.top()]>=belep[v]))
      component[komponensek].add(verem.top())
      vermen[verem.top()]=FALSE
      verem.pop()
```

Egy példa[3]

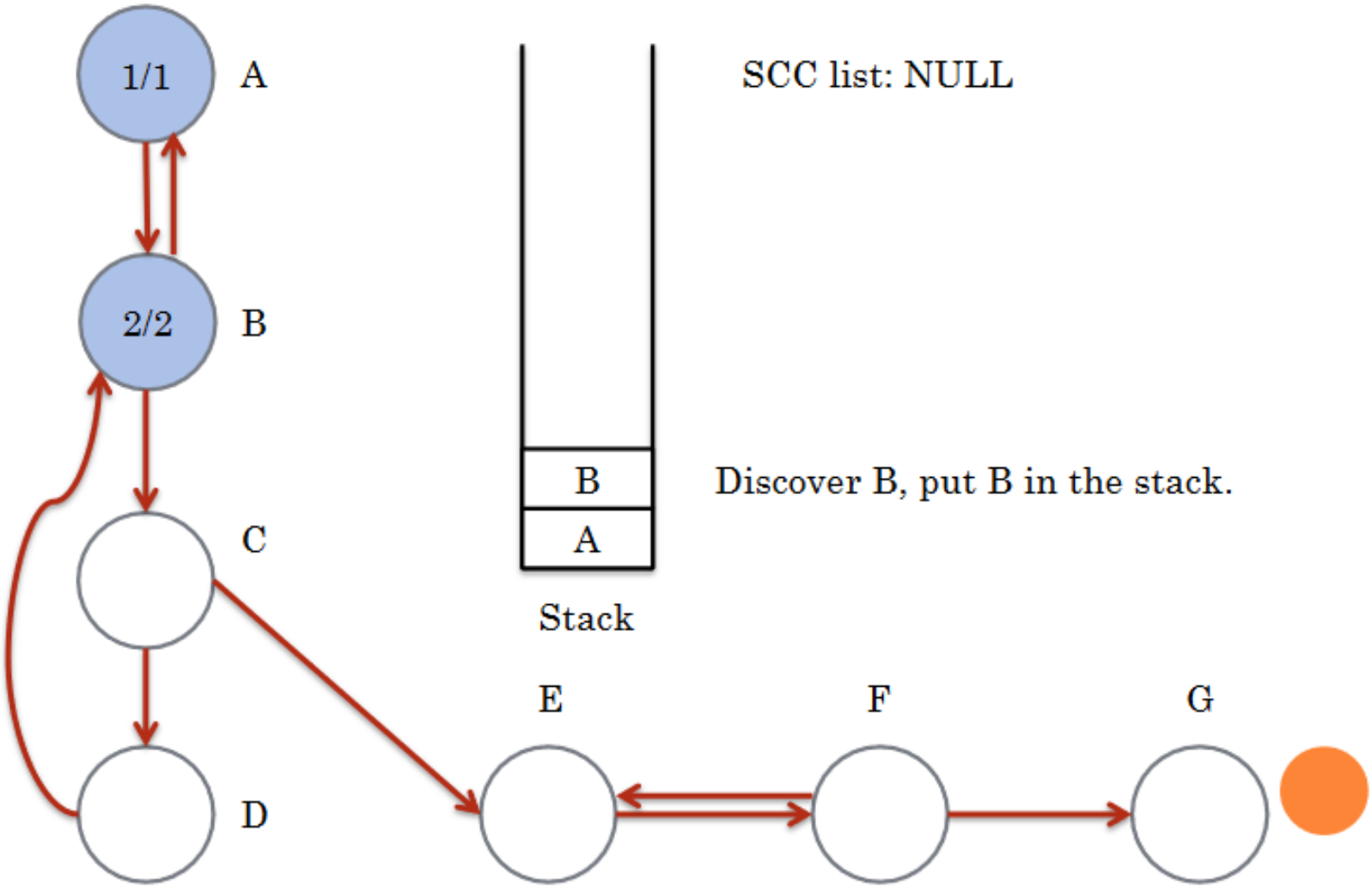


[3] forrás: https://nanopdf.com/download/scc-algorithm-wordpresscom_pdf

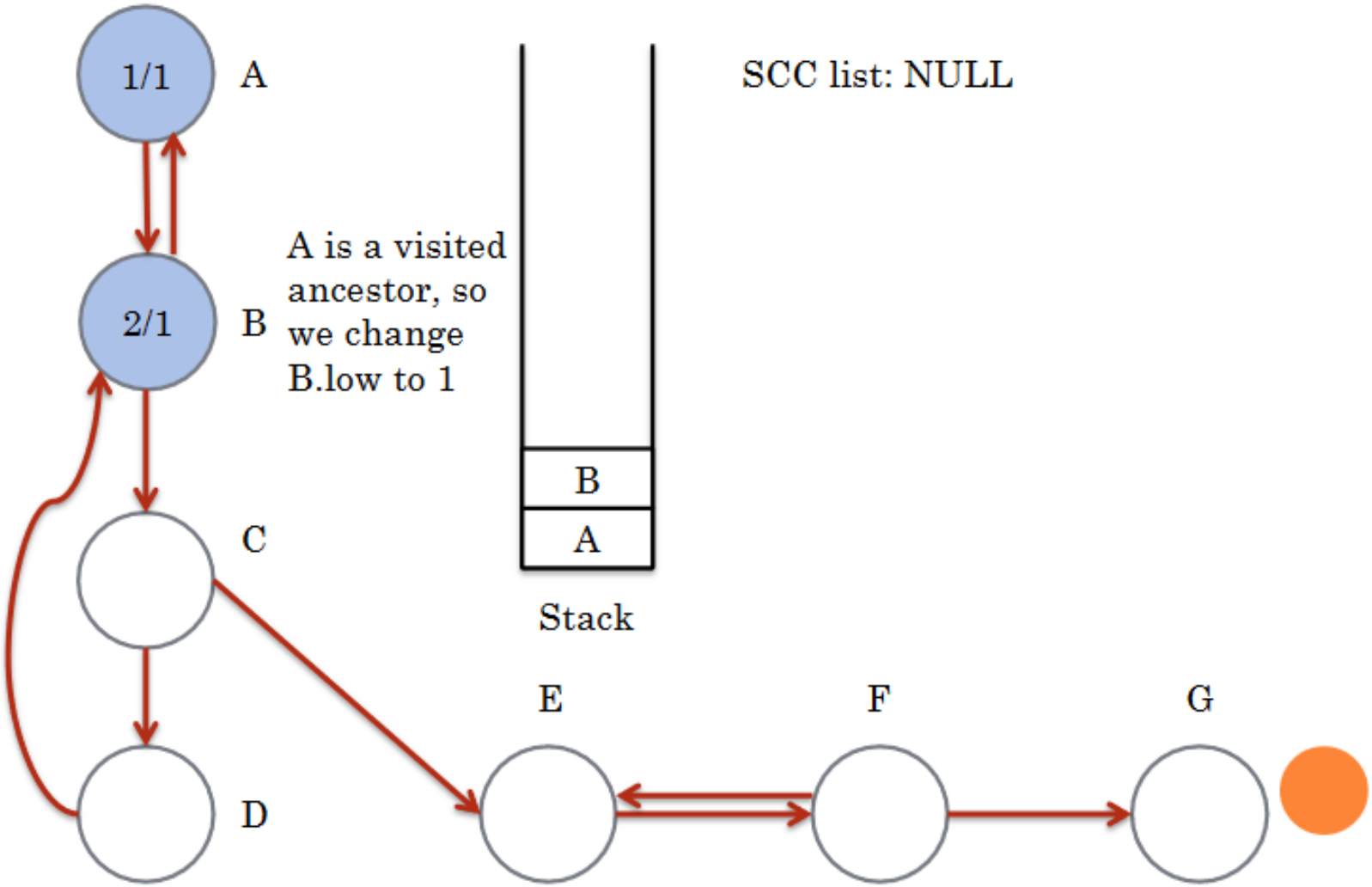
Egy példa (folyt.)



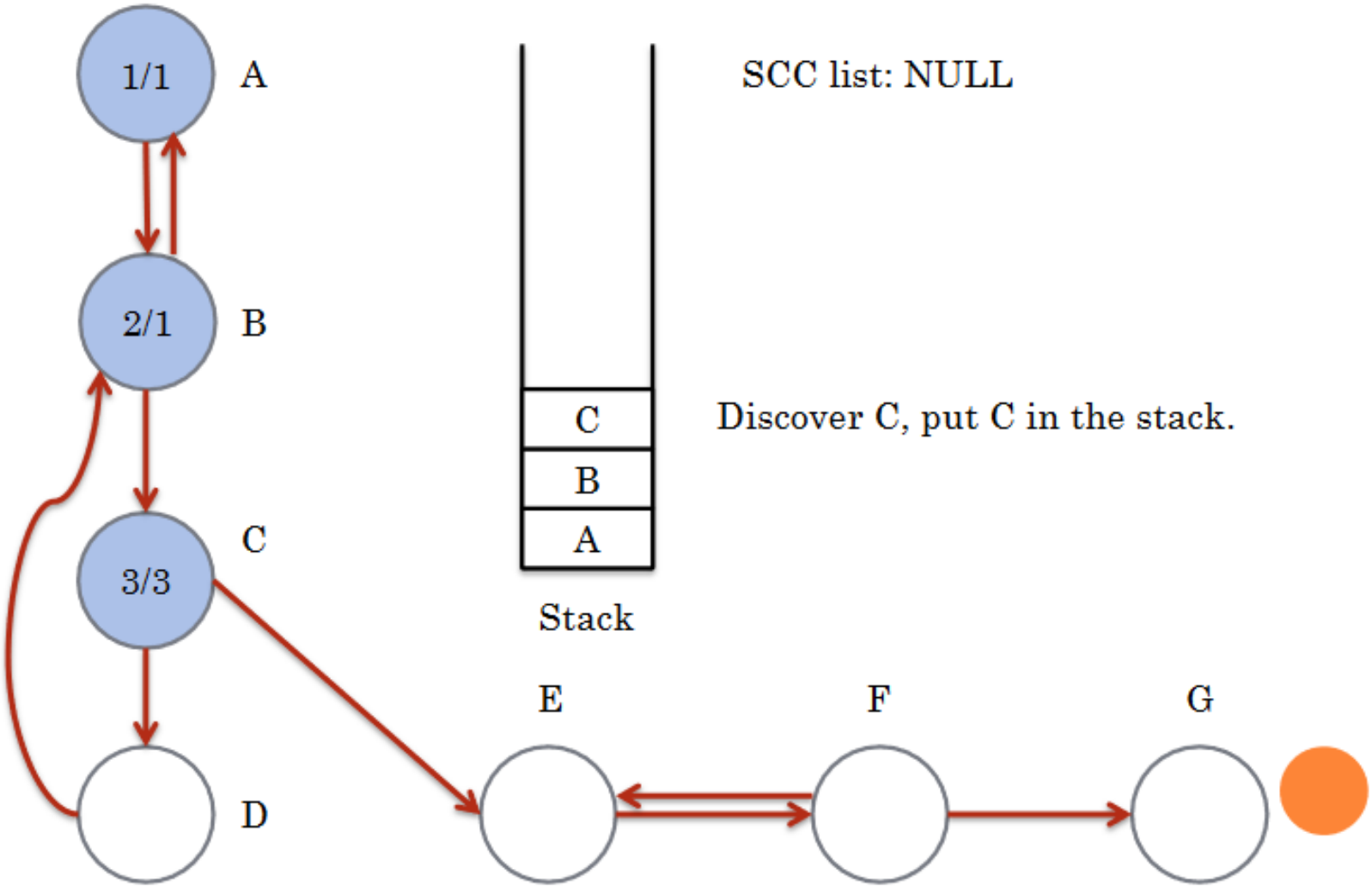
Egy példa (folyt.)



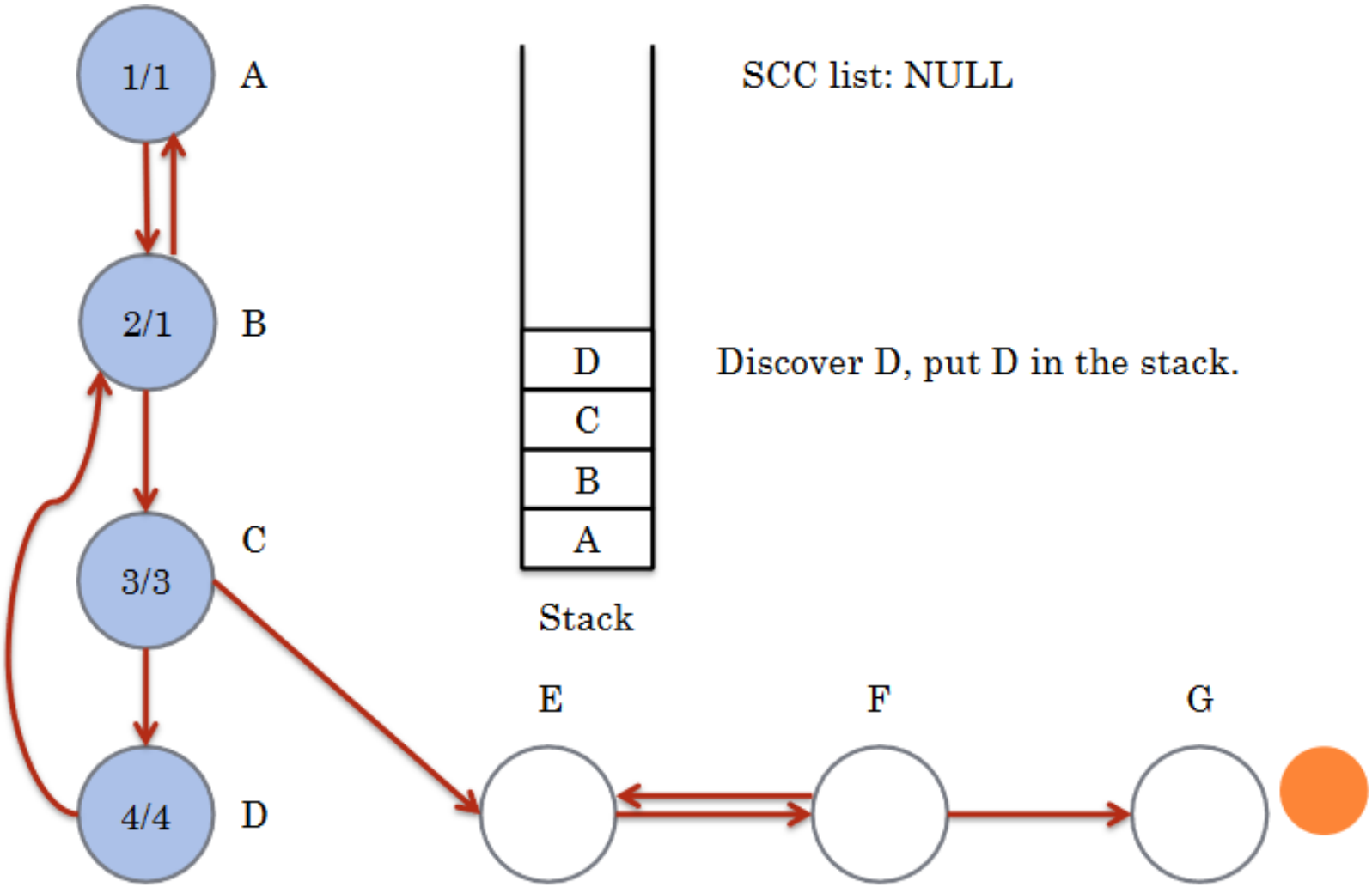
Egy példa (folyt.)



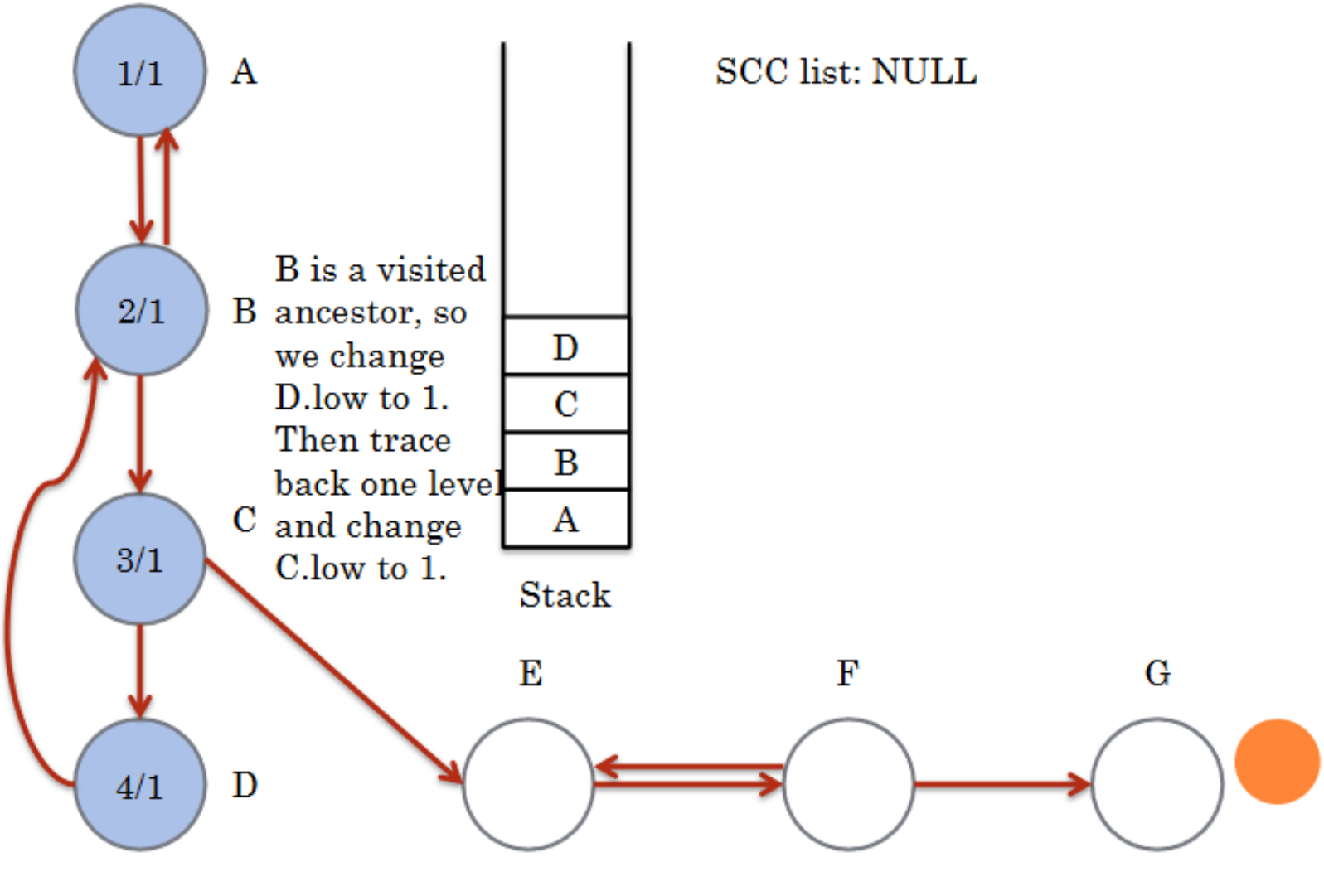
Egy példa (folyt.)



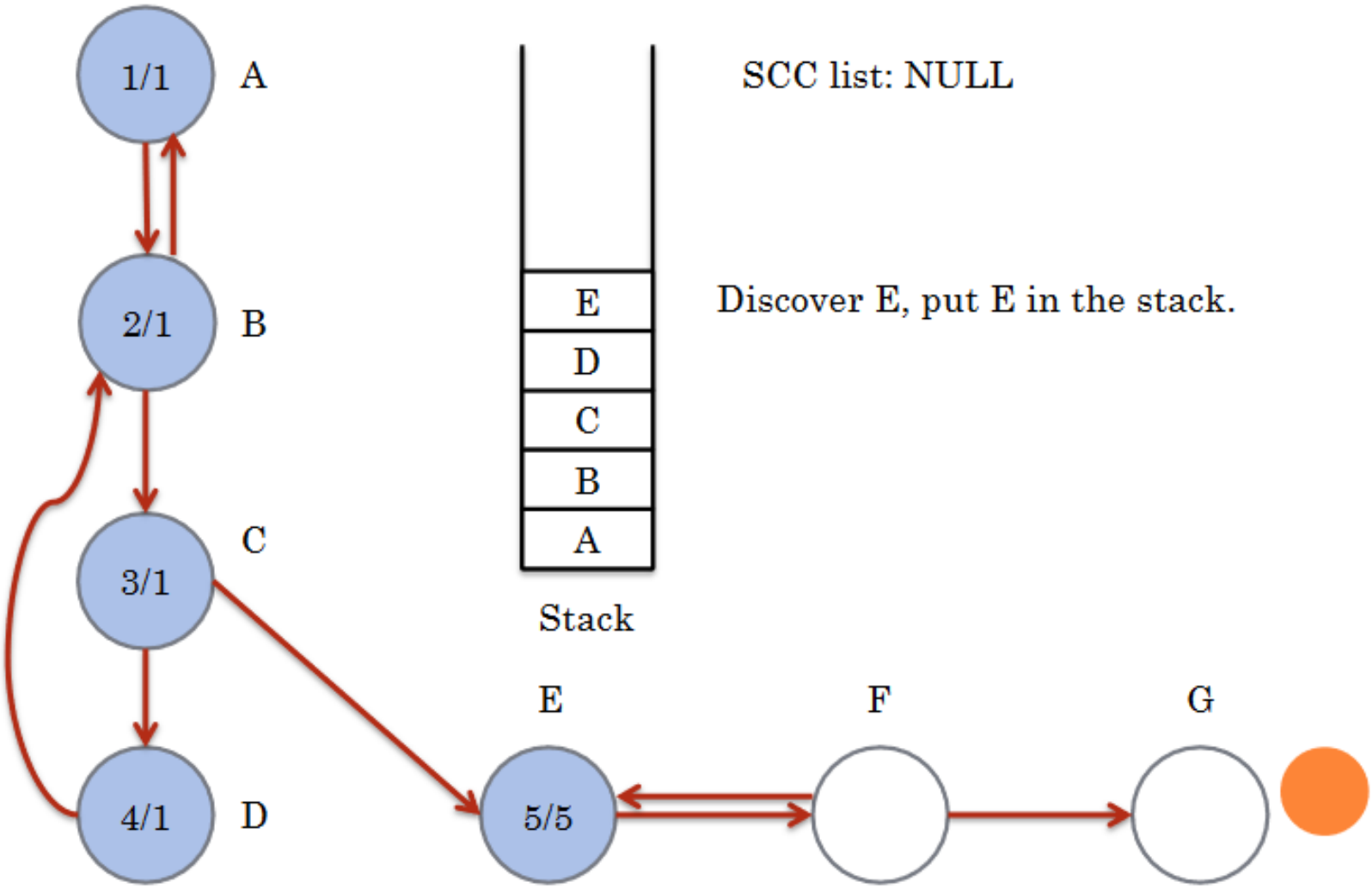
Egy példa (folyt.)



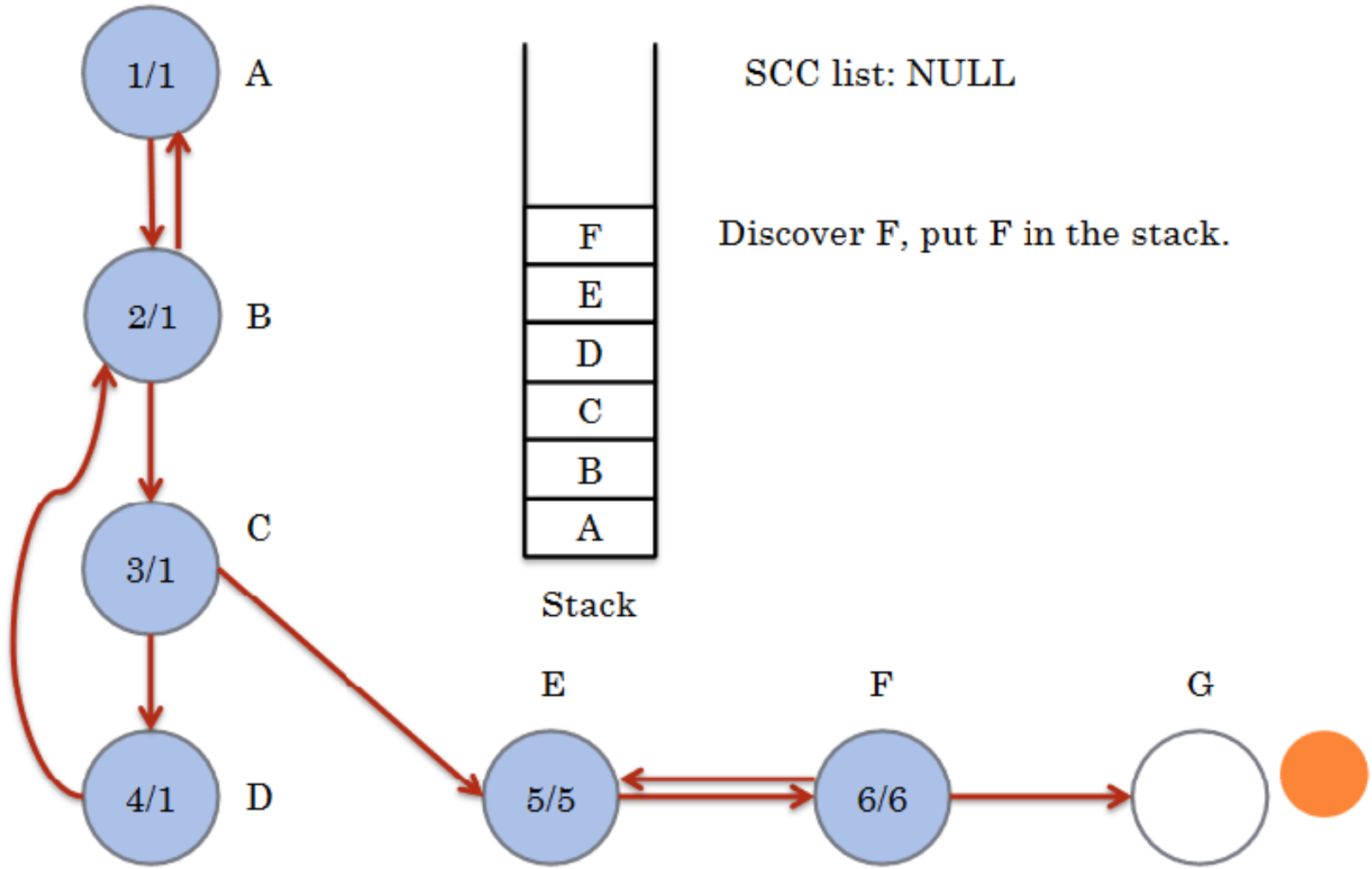
Egy példa (folyt.)



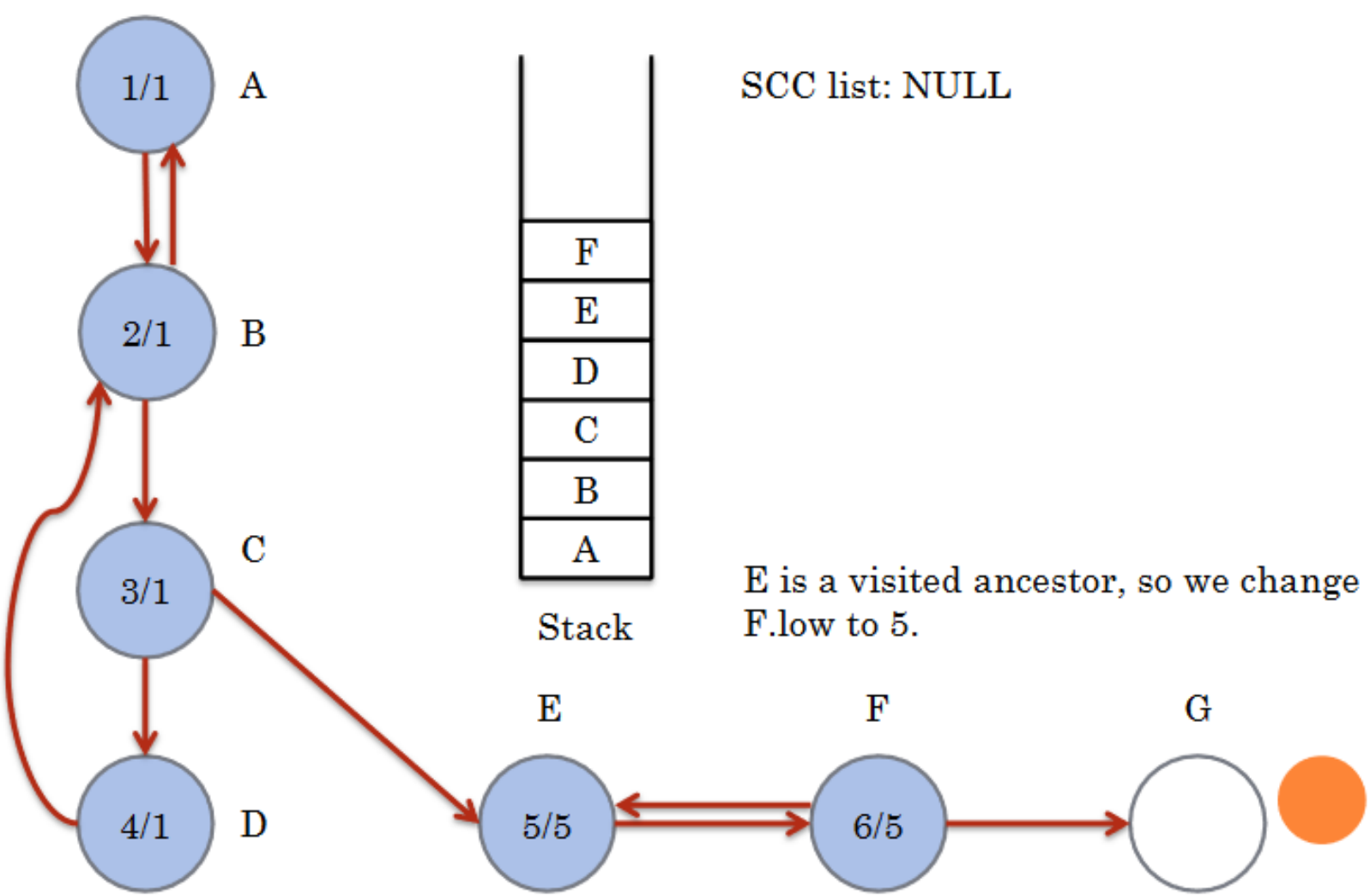
Egy példa (folyt.)



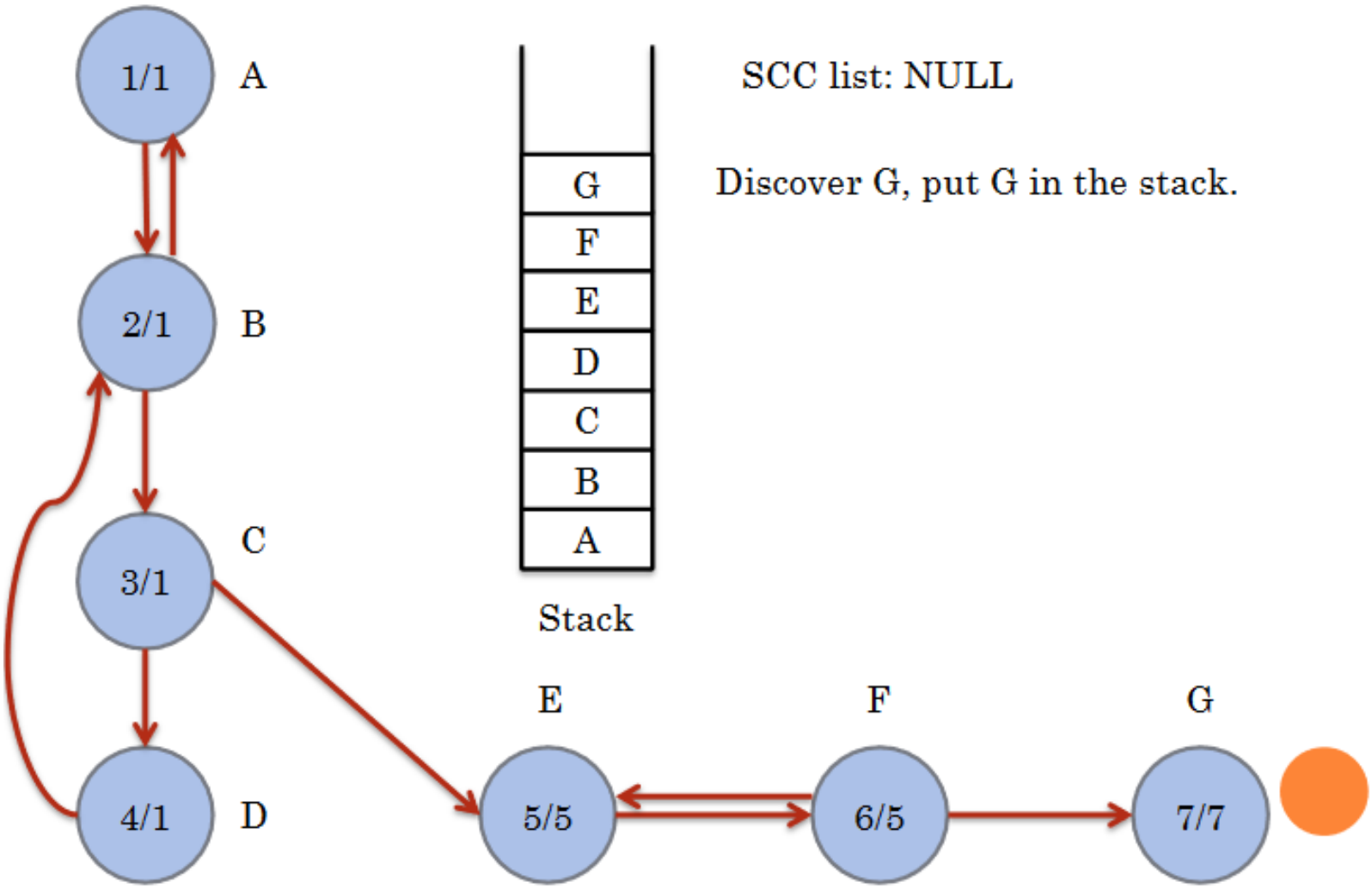
Egy példa (folyt.)



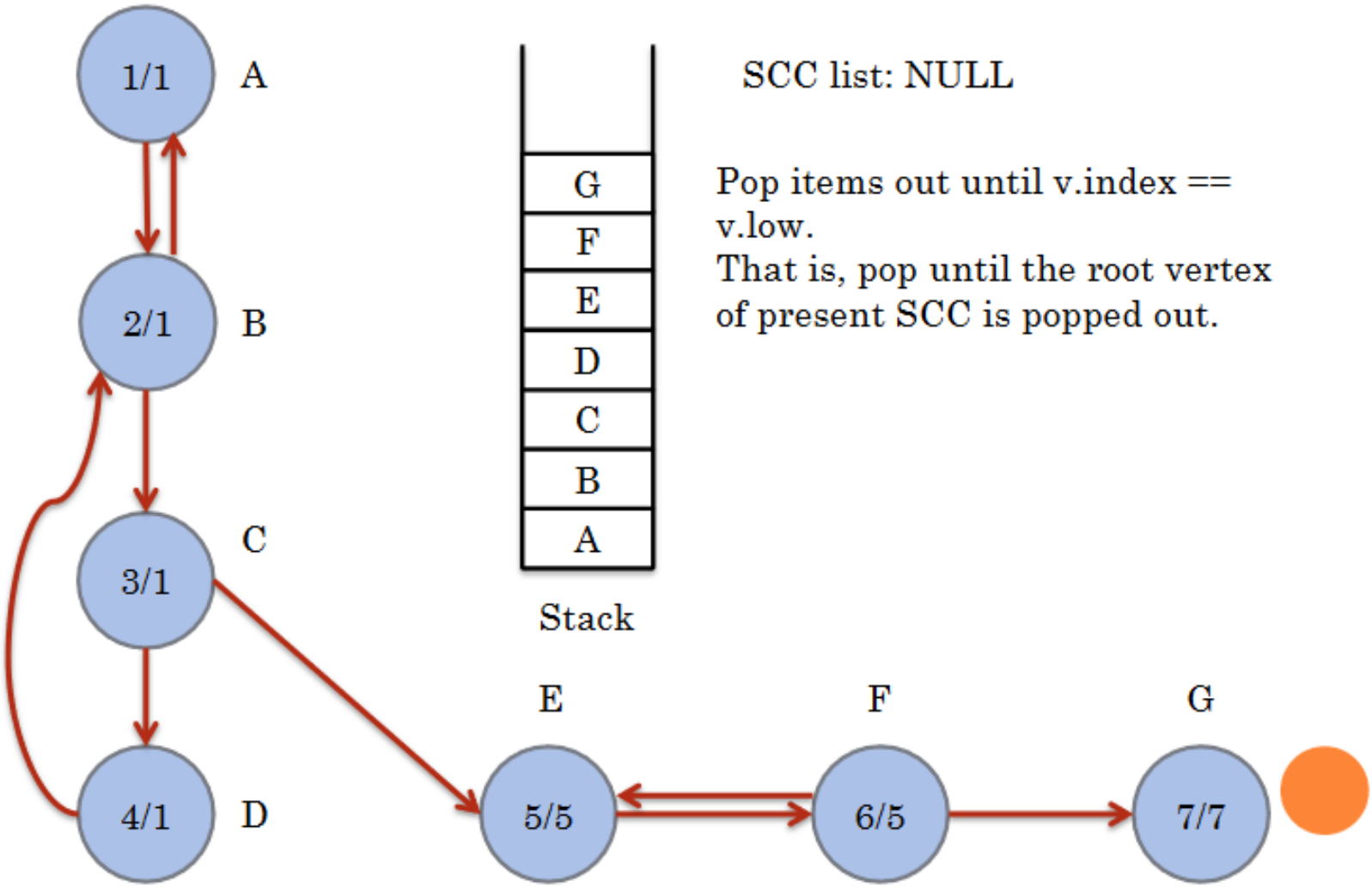
Egy példa (folyt.)



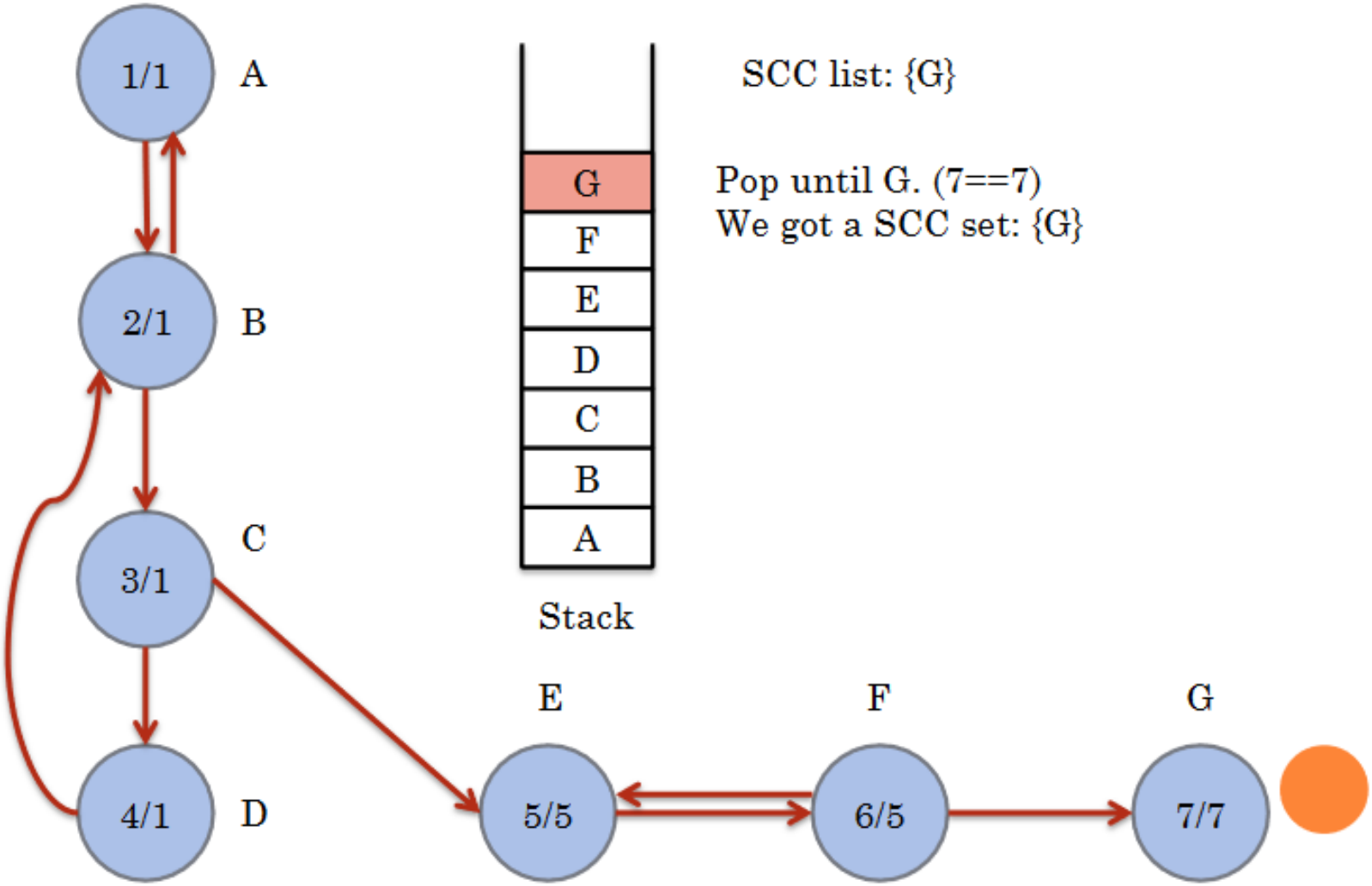
Egy példa (folyt.)



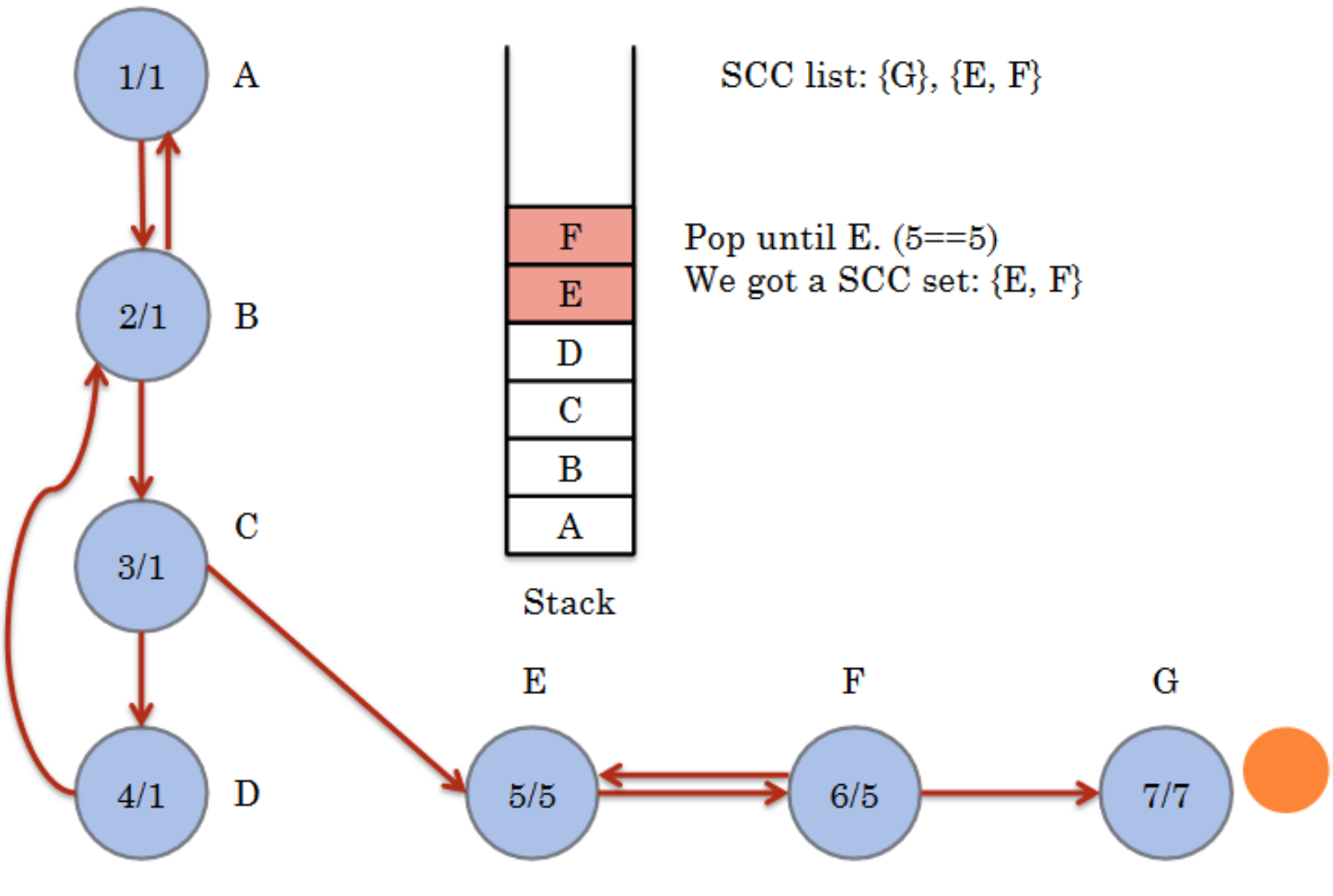
Egy példa (folyt.)



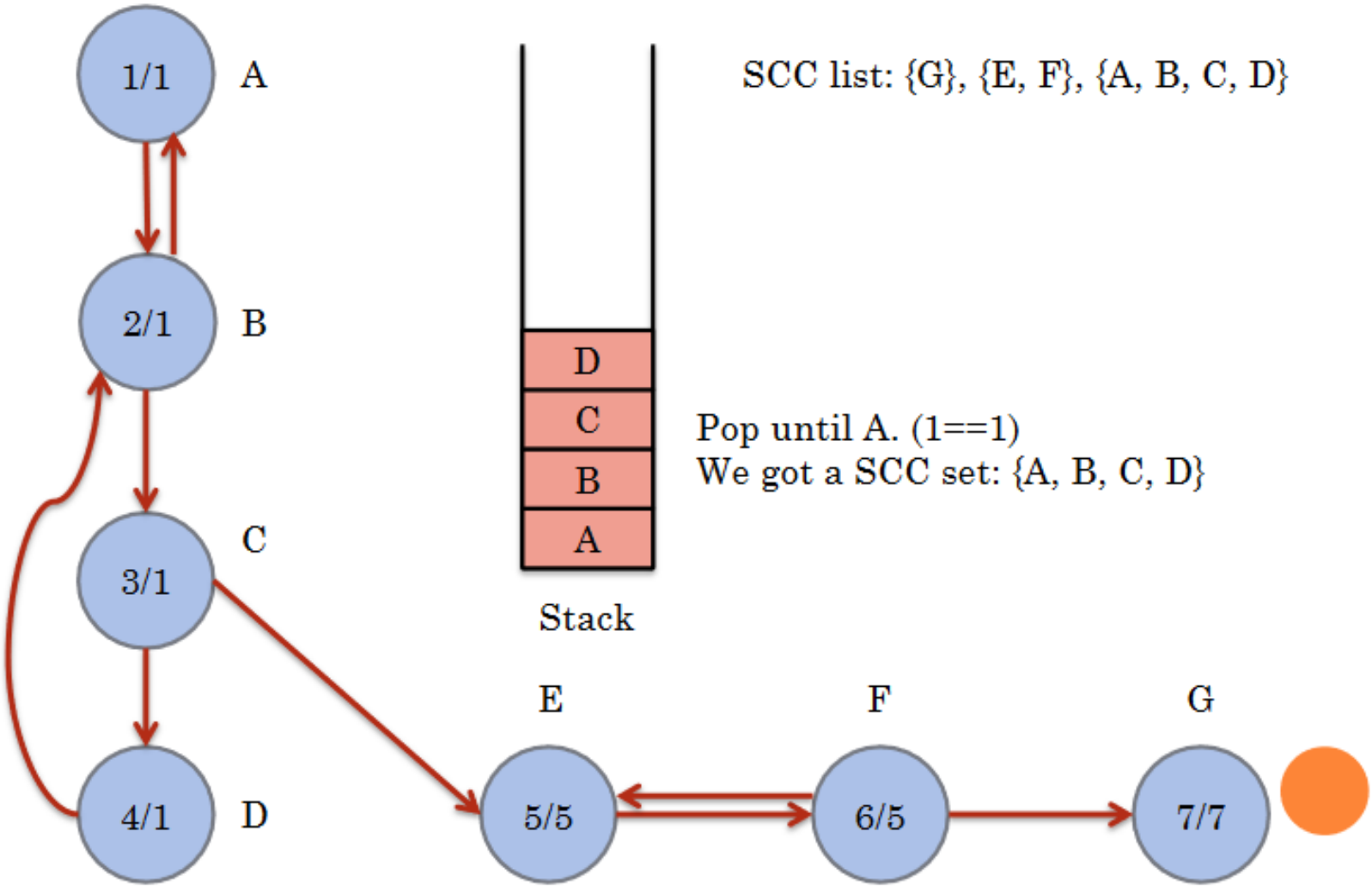
Egy példa (folyt.)



Egy példa (folyt.)



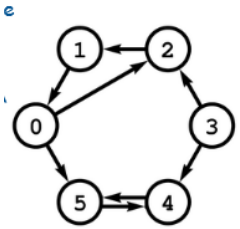
Egy példa (folyt.)



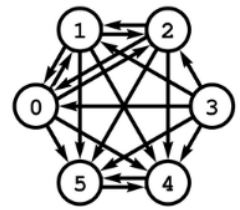
Tranzitív lezárás

Tranzitív lezárás

Egy $G = (V, E)$ gráf tranzitív lezártja (tranzitív lezárása) az a $G' = (V, E')$ gráf, melyben akkor létezik egy (u, v) él, ha a G-ben létezik út u-ból v-be.



	0	1	2	3	4	5
0	1	0	1	0	0	1
1	1	1	0	0	0	0
2	0	1	1	0	0	0
3	0	0	1	1	1	0
4	0	0	0	0	1	1
5	0	0	0	0	1	1



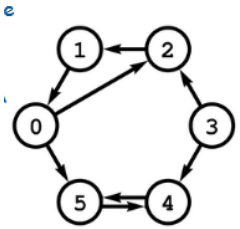
	0	1	2	3	4	5
0	1	1	1	0	1	1
1	1	1	1	0	1	1
2	1	1	1	0	1	1
3	1	1	1	1	1	1
4	0	0	0	0	1	1
5	0	0	0	0	1	1

Megoldás: Warshall algoritmus (következő órákon), vagy

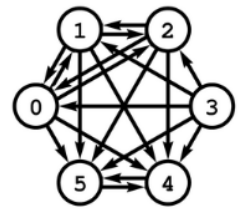
Tranzitív lezárás

Tranzitív lezárás

Egy $G = (V, E)$ gráf tranzitív lezártja (tranzitív lezárása) az a $G' = (V, E')$ gráf, melyben akkor létezik egy (u, v) él, ha a G-ben létezik út u-ból v-be.



	0	1	2	3	4	5
0	1	0	1	0	0	1
1	1	1	0	0	0	0
2	0	1	1	0	0	0
3	0	0	1	1	1	0
4	0	0	0	0	1	1
5	0	0	0	0	1	1



	0	1	2	3	4	5
0	1	1	1	0	1	1
1	1	1	1	0	1	1
2	1	1	1	0	1	1
3	1	1	1	1	1	1
4	0	0	0	0	1	1
5	0	0	0	0	1	1

Megoldás: Warshall algoritmus (következő órákon), vagy DFS

Topológiai rendezés

- DFS-t használva egy kör nélküli irányított gráfot topológiailag rendezhetünk
- egy lineáris elrendezése a csomópontoknak az élek függvényében

Topológiai rendezés

Legyen egy $G = (V, E)$ körmentes gráf, a topológiai rendezés a csomópontok sorát adja meg, úgy hogy ha G tartalmazza az $\{u, v\}$ élt, akkor u hamarabb jelenik meg a felsorolásban, mint v .

- sok helyen használják a topológiai rendezést

Topológiai rendezés (II)

- egy sorrendje bizonyos cselekményeknek, amit egy meghatározott sorrendben kell végrehajtani
- egyes cselekményeket végre kell hajtani, mielőtt mások elkezdődnének
- *milyen sorrendben kell végrehajtani?*
- reprezentálhatjuk ezeket mint egy gráfot, ahol a csomópontok a cselekmények
- az $\{u, v\}$ él azt jelenti, hogy az u cselekmény megelőzi a v-t
- topológiai rendezés során megkapjuk az események végrehajtásának sorrendjét

Topológiai rendezés (III)

Topológiai_rendezés(G)
meghívjuk a DFS(G)-t, hogy meghatározzuk a $v.f, v \in V$-ket
csökkenő sorrendbe rendezzük a $v.f$ szerint (miután a csúcsok befejeződtek beszúrjuk egy listába)
return lista

- a topológiai rendezés ideje $\Theta(V + E)$
- DFS ideje $\Theta(V + E)$
- ahhoz, hogy egy $v \in V$ csúcsot beszúrjunk a listába $O(1)$ időre van szükség

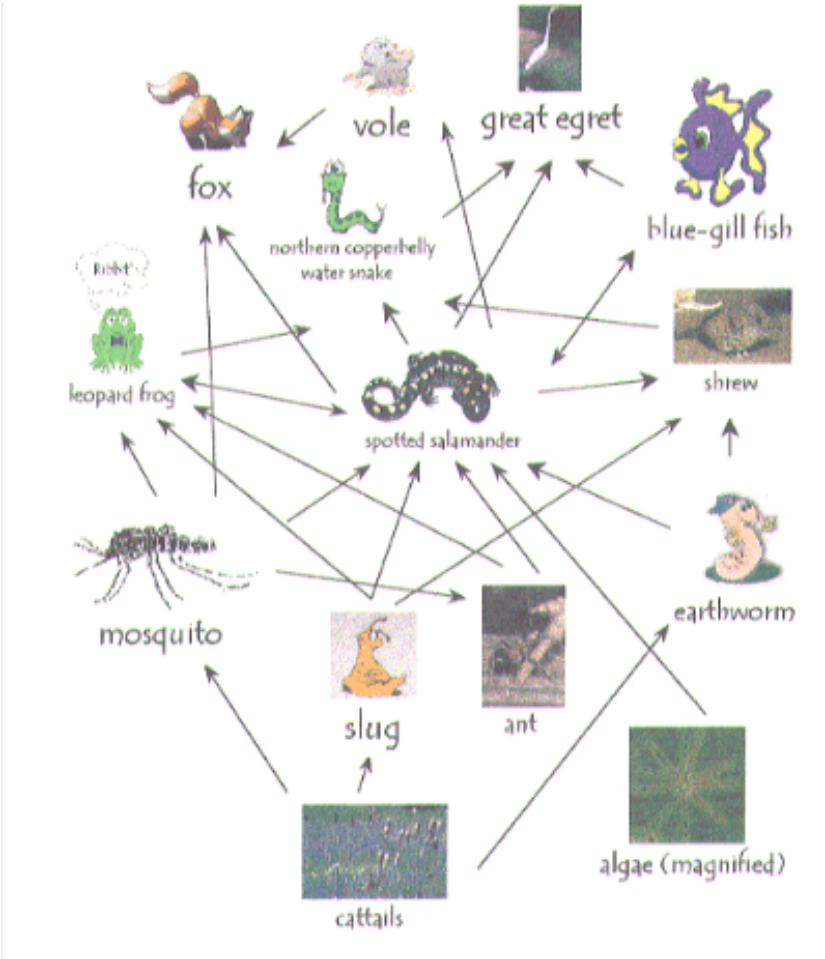
Példa - lásd 2_jegyzet.pdf

Alkalmazások - szélességi és mélységi bejárásra

- szoftverek moduljainak a függősége

Alkalmazások - szélességi és mélységi bejárásra

- szoftverek moduljainak a függősége
- biológia



Alkalmazások (folyt.) - hálózatelemzés

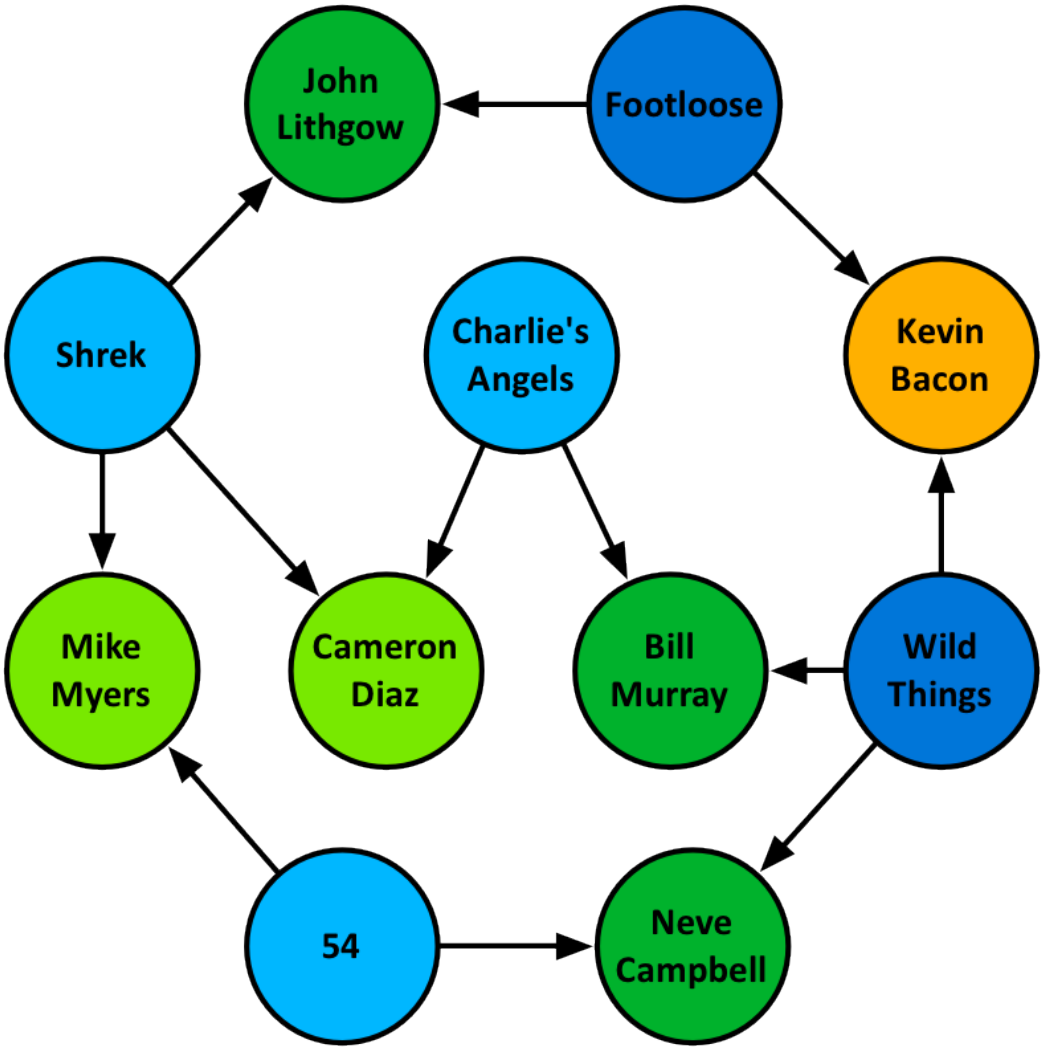
Szélességi bejárásra:

- közösségi hálókban az ismerősök ajánlása (pl. Facebook)

Alkalmazások (folyt.) - hálózatelemzés

Szélességi bejárásra:

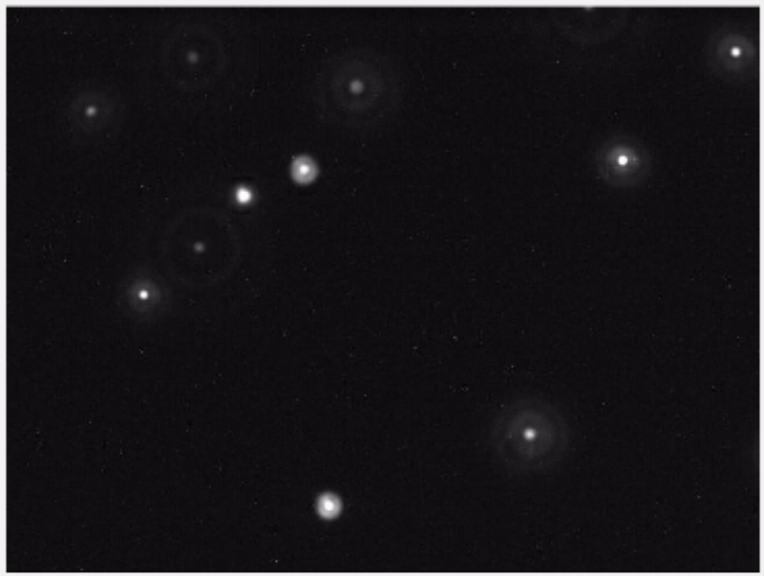
- közösségi hálókban az ismerősök ajánlása (pl. Facebook)
- Kevin Bacon/ Erdős Pál/ Sabbath szám

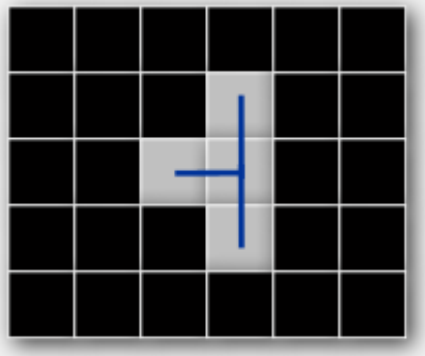

John Lithgow
Footloose
Shrek
Charlie's Angels
Kevin Bacon
Mike Myers
Cameron Diaz
Bill Murray
Wild Things
54
Neve Campbell


Kevin Bacon (I)
Jake Szymanski
Juan Maria Jr.
Deanna Raphael
Danny Mastrogiorgio
Terry Kinney
Rick Hess (I)
Salvatore Paul Piro
Jonathan Tucker (I)
Maria Deasy
Peter Rini
Dash Mihok
Gerrit van der Meer
Mary B. McCann
Billy Malone (II)
Nathan Fillion
Conrad Mcertins Jr.
Ruth Malecczech
Tom Signorelli
Sleepers
Barry Squitieri
Chris Chrisafis
Brooke Devereaux
Mick O'Rourke (I)
Belknap
Monica Polito
Laurie Gould
Gabriel Geer
Kevin Will
Wendell Pierce
Gaby Glatzer
Jay O. Sanders
Barry Levinson (I)
Lisa Lewis (I)
Sean Patrick Reilly
Lennie Loftin
John Murray (II)
Isabel Halliburton
Sturgis Wa
Patrick Tull
Daniel Stern (I)
Ed Helms (I)
Larry Nazimek
George Vincent (I)
Charles Kimbrough
Sol Schwade
Richard Matheson
Thomas James Riscili
Lusia Strus
Angela Rago (I)
Pat McNamara (I)
Connie Fleming
Lorenzo Carcaterra
Peter Giuliano (I)
Rob Corddry
Molly McDougal
Robert Redford (I)
Mark De Alessandro
Roger Delgado
Michael Belleran
George Ivey
Chalon W
Rose Calola
Harriet Rawlings
Burt Reynolds (I)
James Lentzsch
Joseph Perrino
David Buck
Michael Kaufman (I)
Carmine Parisi
Austin Pendleton
Anthony Sinopoli
Conor O'Farrell (I)
Greg Iles
Ralph Tabakin
Peter Mahoney (IV)
Richard Whiting (I)
James L. Brooks
Miranda Bailey
Geoffrey Wigdor
Frank Medrano
Kevin Richard Sullivan
Michael P. Moran (I)
Gina Menza
Bruno Kirby
Peter McRobbie
Gayle Scott (I)
Rocco Musacchia
Larry Romano
Bobcat Goldthwait
Mort Marshall
Eric Geier
Kylie Bax
Victoria Bass
Kevin Scott Greer
Abdullah Bey
Kathryn Erbe
Nick Holmes (I)
Jay Weisleder
Mike White
Everett Fitzgerald
Chais Dean
Fran Rubel Kuzui
Phillip O'Rourke
Dani Sexton (I)
Nancy Duerr

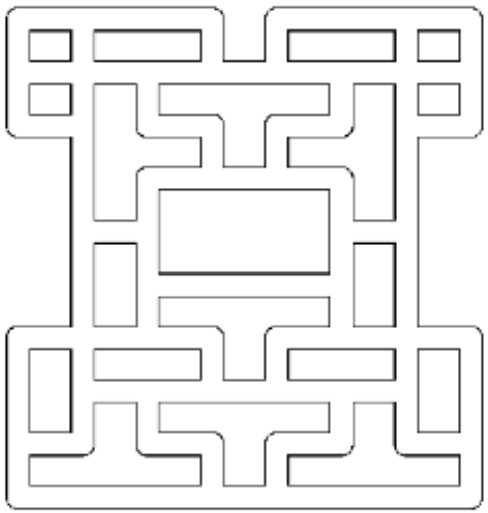
Alkalmazások (folyt.) - képfeldolgozás

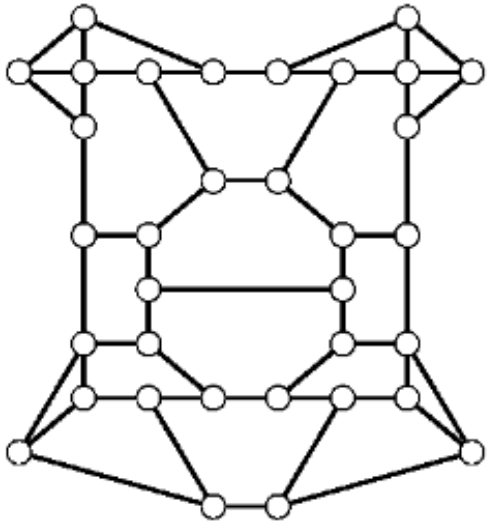
Keressünk nagyobb csillagokat az alábbi képen:





Alkalmazások (folyt.) - Labirintus





Megoldás: Thremaux algoritmusa - a 19. századból, DFS-en alapul

- jegyezd meg minden csomópontot amiben jártál
- jegyezz meg minden bejárt utat

Alkalmazások (folyt.)

- páros gráf ellenőrzés
- kör meghatározása