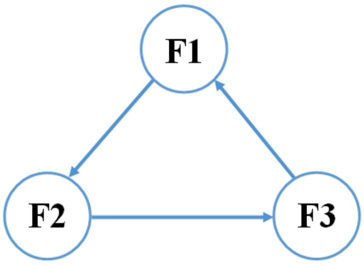
29. DAG TOPOLOGIKUS RENDEZÉSE

A címben szereplő DAG (mint egy angol eredetű mozaikszó) körmentes irányított gráfot jelent.

Operációs rendszerekben, adatbázis kezelő rendszerekben előfordulnak olyan esetek, hogy egyes folyamatok más folyamatokra várnak. Ilyen eset lehet egy nyomtatás várakoztatása, mert a nyomtatót egy másik folyamat használja, vagy egy adattáblán való művelet elvégzésének a várakoztatása, mert egy másik folyamat a táblát zárolta.

Építsünk fel egy úgy nevezett várakozási gráfot, ahol a gráf csúcsai a folyamatok, és egy u csúcsból vezessen él egy v csúcsba, ha az u folyamat a v folyamatra várakozik. Az említett várakozási reláció nem feltétlen szimmetrikus, tehát a gráfunk legyen irányított.

Ha a körbevárakozás esete lép fel, akkor a gráfban irányított kör keletkezik. Ha nem avatkoznánk közbe, akkor a folyamatok elvben akár végtelen ideig is várakozhatnának egymásra. A jelenséget a szakirodalom holtpontnak nevezi. Például a 29.1. ábrán az F1 folyamat várakozik F2-re, F2 várakozik F3-ra, és F3 várakozik F1-re.



29.1. ábra. Egymásra várakozó folyamatok gráfban ábrázolva

Az irányított gráfok körmentessége (illetve, mint hibajelenség, a körök megjelenése) további jelentős szerepet kap az irányított gráfokkal leírható összetett folyamatok „linearizálása" terén. Például, az ételek elkészítése során bizonyos lépéseket előzetesen végrehajtandó tevékenységekhez, korábbi fázisok befejeződéséhez kötnek a receptek. Ezeket a függőségeket egy irányított gráffal lehet ábrázolni, amely – helyes recept esetén – nem tartalmaz kört. Komolyabb kiterjedésű gráfra példa lehet az autógyártásban, a szerelőcsarnok egy pontján elvégzett műveletek egymásutánjának a rendszere.

Mindkét példában fellép az az a feladat, hogy a gráf csúcsaiban szereplő tevékenységeket rendezzük szekvenciába úgy, hogy mire a sor hozzájuk ér, addigra már minden előzetesen végrehajtandó tevékenységen túljutottunk. A 29.2. ábrán szereplő gráf esetén egy ilyen felsorolás lehet például a következő: 1, 3, 2, 5, 4, 6. A csúcsoknak az ilyen alkalmas sorrendbe állítását nevezzük a körmentes irányított gráf topologikus rendezésének.

29.1. A topologikus rendezés jellemzői

Az egymásra épülő tevékenységeket akkor tudjuk végrehajtási sorba rendezni, ha a folyamatok egy *irányított körmentes gráfban* (*directed acyclic graph, DAG*) ábrázolhatók. Ennek megfelelően a fő feladatunk ellenőrizni, hogy irányított gráfunk teljesíti-e a körmentesség feltételét, másként a *DAG tulajdonságot*. A topologikus rendezés azután végezhető el.

A DAG tulajdonság ellenőrzése tehát nem más, mint irányított körök felderítése a gráfban. Ez a feladat pedig visszavezethető a 28. fejezetben ismertetett élosztályozásra.

Például az előző fejezet 28.9. ábráján követhető az a lépés, amelyben a mélységi bejárás a gráfon detektált egy körre utaló visszaélt ($(3, 6)$ él).

Alább megfogalmazunk néhány olyan állítást, amelyek a körmentesség, a mélységi bejárás és a topologikus rendezés között teremtenek kapcsolatot.

Állítás. Legyen $G = (V, E)$ irányított, véges gráf. Ha a G gráf mélységi bejárása során találunk visszaélt, akkor G nem DAG.

Bizonyítás. Tegyük fel, hogy $(u, v) \in E$ egy visszaél, ekkor u leszármazottja v-nek a mélységi fában, azaz létezik $v \rightsquigarrow u$ irányított út. Ehhez hozzávéve (u, v) élt $v \rightsquigarrow u \rightarrow v$ egy irányított kör.

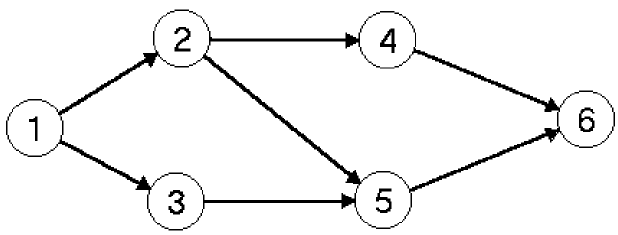
Állítás. Legyen $G = (V, E)$ irányított, véges gráf. Ha a G gráf nem DAG, akkor minden mélységi bejárás során találunk visszaélt.

Bizonyítás. G nem DAG, azaz van benne irányított kör, legyen $v \rightsquigarrow u \rightarrow v$ egy irányított kör, és legyen v a mélységi bejárás során elsőnek elért csúcs. Ekkor a kör mentén a v kivételével minden csúcs fehér ebben a pillanatban. A kör mentén, vagy kis kerülő úton, de fehér csúcsok mentén eljutunk v-ből u-ba (ha van $v \rightsquigarrow u$ út, akkor van csupa fehér csúcsból álló $v \rightsquigarrow u$ út is), tehát (u, v) visszaél lesz.

Arra a kedvező eredményre jutottunk, hogy egy gráfról $O(n + e)$ idő alatt eldönthető, hogy DAG-e, a mélységi bejárás felhasználásával. Nem kell mást tenni, mint a mélységi keresés során figyelni az éltípusokat, ha nem találunk visszaélt, akkor a gráf DAG.

Definíció. Legyen $G = (V, E)$ irányított, véges gráf, továbbá legyen $n = |V|$. G csúcsainak egy $v_1, \ldots, v_n$ felsorolása, G egy *topologikus rendezése*, ha $\forall\,(u, w) \in E$ él esetén a felsorolásban u előbb áll, mint w, azaz $u = v_i$ és $w = v_j$ esetén $i < j$.

Például vegyük a 29.2. ábrán látható körmentes gráfot. A gráf egy topologikus rendezése 1, 2, 3, 4, 5, 6 (ez most egy másik rendezés, mint amely előzőleg szerepelt).



29.2. ábra. Egy körmentes irányított gráf

Állítás. Ha $G = (V, E)$ irányított gráf DAG, akkor $\exists\ u \in V$ csúcs, amelybe nem fut él.

Bizonyítás. Indirekt tegyük fel, hogy nem létezik ilyen csúcs. Ekkor vegyünk egy csúcsot, és egyik befutó élén hátráljunk (lépjünk vissza a megelőző csúcsra). Azonban minden csúcsnak van befutó éle, így a hátrálást korlátlanul végrehajthatjuk, vagyis kört kapunk, ami ellentmondás.

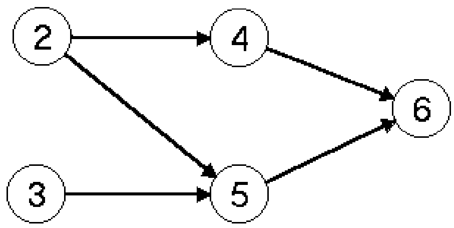
Állítás. A $G = (V, E)$ gráfnak létezik topologikus rendezése akkor, és csak akkor, ha G DAG.

Bizonyítás.

$\Longrightarrow$: G-nek létezik topologikus rendezése, és indirekt tegyük fel, hogy G nem DAG, azaz van benne irányított kör, ami ellentmondás, mivel a kör mentén nem lehet alkalmas sorrendet definiálni.

⟸: Konstruktív bizonyítást adunk a csúcsok száma szerinti teljes indukcióval. Az egyetlen pontból álló gráfra utaló $n = 1$ esetén az állítás triviális módon igaz. Tegyük fel, hogy $n - 1$ pontú DAG-okra igaz az állítás. Legyen $G_n = (V, E)$ csúcsból álló DAG, ekkor $\exists\, u \in V$ csúcs, amelyben nem megy él. Töröljük a gráfból u-t és a belőle kimenő éleket, így megkapjuk G_{n-1} $n - 1$ csúcsból álló DAG-ot. Ekkor az indukciós feltevés szerint G_{n-1}-nek létezik topologikus rendezése, legyen ez $v_1, \dots, v_{n-1}$. Ekkor $u, v_1, \dots, v_{n-1}$ egy topologikus rendezése lesz G_n-nek.

Például legyen G_n a 28.2. ábra gráfja. Ekkor az 1-es csúcs törlésével megkapjuk a G_{n-1} gráfot (28.3. ábra), ami az indukciós feltétel szerint DAG, így létezik topologikus rendezése: 2,3,4,5,6. Tehát G_n-nek topologikus rendezése az 1,2,3,4,5,6 sorozat, mivel nem megy él az 1-es csúcsba, így az 1-es csúcsot tehetjük a sorozat elejére.



29.3. ábra. Az 1-es csúcs és éleinek elhagyása a gráfból

29.2. A topologikus rendezés algoritmusa

A topologikus rendezés előállítását többféle módon is megközelíthetjük.

- Az előző állítás bizonyításában használt indukció megad egy algoritmust. Legyen Q egy sort adatszerkezet, kezdetben üres.
  1. Q-ba berakjuk azon csúcsokat, amelybe nem megy él.
  2. Ha Q üres, akkor készen vagyunk.
  3. Vegyük ki Q-ból az első elemet, és írjuk ki, legyen ez u.
  4. Töröljük G-ből $(u, v) \in E$ éleket $\forall v \in V$ csúcsra. Ha v-be most már nem megy él, akkor helyezzük v-t a sor végére.
  5. Folytassuk a 2. lépéssel.

• A visszaélek azonosítására szolgált a mélységi bejárás. Ennek segítségével is dolgozhatunk. Futassuk le a mélységi bejárást a gráfon, majd a csúcsokat írjuk ki a csúcsok befejezési számainak (bszám) csökkenő sorrendjében. Az algoritmusban, amikor egy csúcsot elhagyunk, rakjuk a csúcs címkéjét egy verembe, majd a bejárás befejeztével ürítsük ki vermet. A bejárás a DAG tulajdonságot is ellenőrzi.

Állítás. Az eljárás $G = (V, E)$ DAG egy topologikus rendezését állítja elő.

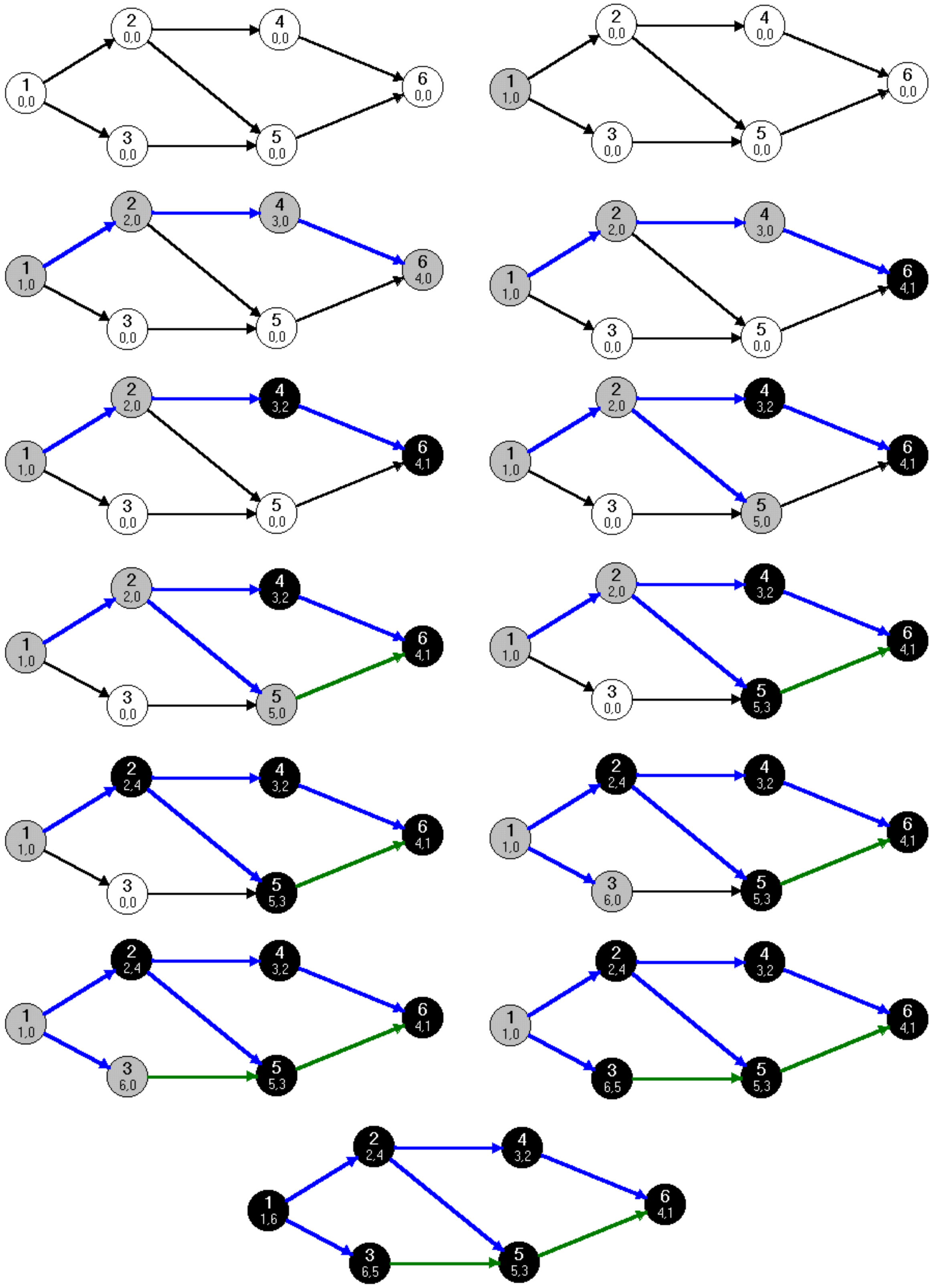
Bizonyítás. Azt kell belátni, hogy amennyiben $(u, v) \in E$, akkor $\text{bszám}[u] > \text{bszám}[v]$. Amikor (u, v) élt vizsgáljuk, akkor v fehér vagy fekete (szürke esetén visszaél lenne, de G DAG). Ha v fehér, akkor v leszármazottja u-nak, tehát $\text{bszám}[u] > \text{bszám}[v]$. Ha v fekete, akkor v-t már megvizsgáltuk és elhagytuk, míg u-t még nem, tehát $\text{bszám}[u] > \text{bszám}[v]$. Azaz minden esetre fennáll az állítás.

Mivel a mélységi bejárás műveletigénye $O(n + e)$, és a veremműveletek száma a csúcsok számával arányos, azaz $\Theta(n)$, $T(n) = O(n + e)$.

29.3. A topologikus rendezés szemléltetése

Nézzük meg a mélységi bejárás alapú algoritmus működését a 29.4. ábrán.

A mélységi bejárással elindulunk az 1-es csúcsból, majd néhány lépés után eljutunk a 6-os csúcsba, miközben csupa faéleket azonosítunk.



29.4. ábra. A topologikus rendezés végrehajtása

Elsőként a 6-os csúcs bejárását fejezzük be, tehát a 6-ost bedobjuk a V verembe, így $V = [6]$. A következő lépésben a 4-es csúcsot fejezzük be. $V = [6, 4]$. Majd a 2-es csúcsból ellátogatunk az 5-ös csúcsba, miközben a verem változatlan marad. Az 5-ös csúcsból kimenő élt keresztélként azonosítjuk. Befejezzük az 5-ös csúcs bejárását is, tehát a verembe dobjuk: $V = [6, 4, 5]$. Következőnek befejezzük a 2-es csúcsot is, és a vermet a 2-essel bővítjük: $V = [6, 4, 5, 2]$. Az 1-es csúcsból elérjük a 3-as csúcsot, és újabb keresztélt azonosítunk: (3,5). Befejezzük a 3-as csúcs bejárását is. Befejezzük a 3-as csúcs bejárását is. $V = [6, 4, 5, 2, 3]$. Végezetül a bejárás utolsó csúcsaként elhagyjuk az 1-es csúcsot is. A veremben van a gráf összes csúcsa a befejezésük szerint: $V = [6, 4, 5, 2, 3, 1]$. Menet közben nem találtunk visszaélt, tehát a DAG tulajdonságot rendben találtuk. Végül a verem tartalmát kiírjuk: 1,3,2,5,4,6, amellyel megkapjuk a G egy topologikus rendezését.