

Debreceni Egyetem
Természettudományi Kar
Matematikai Intézet

Szakdolgozat

Gráfok címkézéseinek vizsgálata

készítette:

Juhász Gyula

témavezető: Dr. Tengely Szabolcs

Debrecen, 2016

TARTALOMJEGYZÉK

Tartalomjegyzék	i
1 Bevezető	3
2 Alapvető definíciók és tételek	5
2.1. A gráfok, fa gráfok és páros gráfok	5
3 Graceful gráfok és graceful címkézés	11
3.1. Graceful címkézés, illetve címkézés teljes gráfok és fák esetén . . .	11
4 Különleges gráfok, konstrukciók és Sage implementáció	19
4.1. Graceful double-wheel gráf konstrukciója	19
4.2. Graceful caterpillar gráf konstrukciója	30
4.3. Graceful n-szeresen összetett teljes páros gráf konstrukciója	32
4.4. Graceful pendant gráf konstrukciója	37
Irodalomjegyzék	47

KÖSZÖNETNYILVÁNÍTÁS

Köszönetet szeretnék mondani Dr. Tengely Szabolcs témavezetőmnek, akinek segítségével nem jöhetett volna létre ezen dolgozat. Valamint köszönettel tartozom családomnak és barátaimnak, akik mindig mellettem voltak és támogattak.

1. BEVEZETŐ

A matematikában, azon belül a gráfelméletben a mai napig még mindig rengeteg a megválaszolandó kérdés. Ilyen kérdésekre bukkanunk, hogyha el kezdjük vizsgálni a gráfok címkézéseit, főleg a 20. század egyik címkézési problémáját, a graceful címkézést.

Ezeket a gráfokat már több mint 40 éve vizsgálják, amióta 1967-ben Alexander Rosa publikálta [11]. Egy gráfot gracefulnak nevezünk, ha a csúcsait fel tudjuk úgy címkézni egész $\{0, \dots, e\}$ értékekkel, hogy minden egyes élnek különböző legyen az az értéke, amit a végpontok abszolút különbségéből számolunk ki.

Ez egy nagyon szép és egyben érdekes címkézési formula, ami felvet néhány kérdést. Vajon minden gráfnak létezik graceful címkézése? Van olyan gráfosztály, amire nem teljesül? Tudunk-e olyan algoritmust adni, amely bizonyos gráfok esetén, a csúcsok növelésével is helytálló, azaz ha egy három csúcsú fa gráfra igaz az állítás, akkor az n csúcsúra is igaz lesz? Ezeket a kérdéseket tanulmányozzák, tanulmányozták a gráfelméletbe, de mire is jó, hogy egy gráfnak létezik graceful címkézése?

A graceful gráfokat főleg a kriptográfiában használják, de a csillagászatban, a kristálytanban vagy a röntgennél is felhasználják őket [5].

Viszont a gráfelmélet ezen részén sokan próbálkoznak új gráfok létrehozásában is, így megismerkedhetünk a double-wheel gráfokkal vagy a caterpillar gráfokkal, amik például úgy keletkeztek, hogy próbálták a fa gráfokról belátni, hogy graceful, de mivel nem tudták belátni, ezért a fa gráfokat el kezdték osztályozni. Így rengetek osztályról tudunk a fák esetén beszélni, például a caterpillar gráfokról, super-caterpillar gráfokról, spider gráfokról vagy lobster gráfokról.

Így a dolgozat célja az, hogy bemutassam a graceful címkézést, a fa gráfok bizonyos osztályait, rájuk vonatkozó tételeit, illetve különleges graceful gráfok címkézéseit.

2. ALAPVETŐ DEFINÍCIÓK ÉS TÉTELEK

A legelső fejezetben néhány alapvető kérdésre szeretnék választ adni, mint például mi az a gráf. Ezenkívül néhány alapvető definíciót, tételt és a bizonyítását is szeretnék ismertetni, amiket a graceful gráfok esetén érdemes tudni, és amik közül használni is fogok egy párat a graceful gráfok ismertetésekor.

2.1. A gráfok, fa gráfok és páros gráfok

2.1.1. Definíció. Legyen $V \neq \emptyset$ és E véges halmazok, $\varphi : E \rightarrow \mathcal{P}_{1,2}(V) = \{A \subseteq V \mid |A| = 1 \text{ vagy } 2\}$.

Ekkor $G = (V, E, \varphi)$ -t irányítatlan gráfnak, V elemeit a G gráf csúcsainak, E elemeit a G gráf éleinek nevezzük. Ha $\varphi(e) = \{v, w\}$, ahol $e \in E$, akkor azt mondjuk, hogy a v és w az e él végpontjai.

Ha $E = \emptyset$, akkor a gráfot üresgráfnak nevezzük.

2.1.2. Definíció. Legyen $G = (V, E, \varphi)$ egy véges gráf. Ha $|\varphi(e)| = 1$, akkor $e \in E$ -t hurokélnek nevezzük. Az $e_1, e_2 \in E$ párhuzamos élek, ha $\varphi(e_1) = \varphi(e_2)$. Ha a gráf nem tartalmaz sem hurokért és párhuzamos éleket sem, akkor a G gráfot egyszerűnek nevezzük.

2.1.3. Definíció. Egy $G = (V, E, \varphi)$ véges gráf, egy $v \in V(G)$ fokszámán a v -re illeszkedő élek számát értjük. Ha a gráfban szerepel hurokél is, azokat a fokszám számolásakor 2-szer számoljuk. Jele: $d(v)$.

2.1.1. Tétel (Kézfogási-tétel). Legyen $G = (V, E, \varphi)$ véges gráf. Ekkor

$$\sum_{v \in V(G)} d(v) = 2 \cdot |E(G)|.$$

2.1.1. *Megjegyzés.* Azaz, a fokszámok összege ugyanannyi, mint az élek számának a duplája.

Bizonyítás. Egy élt a fokszámösszegbe 2-szer számolunk bele, ugyanígy számoljuk mindkét végpontjánál is. Ha a gráf tartalmaz hurokért is, akkor is igaz a tétel. $\square$

1. Következmény. Egy G gráfban a fokszámok összege mindig páros.

2. Következmény. Egy G gráfban a páratlan fokú csúcsok száma páros.

Bizonyítás. Valóban, a $\sum_{v \in V(G)} d(v)$ összeget fel lehet bontani két külön részre, a páros és a páratlan fokú csúcsokra. Azaz,

$$\sum_{v \in V(G)} d(v) = \sum_{v \in V(G) \mid d(v) \equiv 0 \pmod{2}} d(v) + \sum_{v \in V(G) \mid d(v) \equiv 1 \pmod{2}} d(v) = 2 \cdot |E|.$$

Ebből azonnal látszik, hogy a jobb oldalon az első $d(v)$ tagjainak a száma páros, és mivel páratlan sok páratlan szám összege páratlan, ezért a második $d(v)$ tagjainak a száma is csak páros lehet. $\square$

2.1.4. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. Ha a gráf egyszerű és bármely két különböző csúcsa szomszédos egymással, akkor a gráfot teljesnek nevezzük. Jele: K_n .

2.1.2. Tétel. Legyen $G = (V, E, \varphi)$ egy n csúcsú teljes gráf. Ekkor éleinek száma:

$$\frac{n \cdot (n-1)}{2}.$$

Bizonyítás. Egy n csúcsú teljes gráfban minden csúcs fokszáma $n-1$, így a fokszámok összege $n \cdot (n-1)$. Így a kézfogási-tétel szerint az élek száma $\frac{n \cdot (n-1)}{2}$. $\square$

2.1.5. Definíció. Legyen $G = (V, E, \varphi)$ egyszerű, véges gráf. Ekkor G komplementere az a $\overline{G}$ véges, egyszerű gráf, amire az teljesül, hogy $V(\overline{G}) = V(G)$ és $\overline{G}$ -ben két csúcs pontosan akkor szomszédosak, ha G -ben nem szomszédosak.

2.1.2. Megjegyzés. Tehát a $\overline{G}$ és G teljes gráfot alkot együtt.

2.1.6. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = n+1$ és $|E(G)| = n$. Ekkor $(v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n)$ -t sétának nevezzük, ha $v_0, v_1, v_2, \dots, v_{n-1}, v_n \in V$, $e_1, e_2, \dots, e_n \in E$ és $\varphi(e_i) = \{v_{i-1}, v_i\}$, ahol $i = 1, \dots, n$. A séta nyílt, ha $v_0 \neq v_n$, ellenben zárt, ha $v_0 = v_n$. Illetve a séta hossza, a benne szereplő élek száma.

2.1.7. Definíció. Egy sétát vonalnak nevezünk, ha a benne szereplő élek különbözőek. Egy séta út, ha nincs benne ismétlődő csúcs és él. Illetve egy séta kör, ha zárt és nincs benne ismétlődő él és csúcs sem, kivéve a kiinduló és az utolsó csúcs egyezését.

2.1.8. Definíció. Egy $G = (V, E, \varphi)$ véges gráf összefüggő, ha bármely két különböző csúcsa között vezet út.

2.1.9. Definíció. Legyen $G = (V, E, \varphi)$ véges, összefüggő gráf. A $v, w \in V(G)$, $v \neq w$ csúcsok távolságán az őket összekötő út hosszát értjük. Továbbá v és v távolsága legyen 0. Jele: $d(v, w)$.

2.1.3. Tétel. Legyen $G = (V, E, \varphi)$ véges, összefüggő gráf.

$$i \quad \forall v, w \in V(G) : d(v, w) \geq 0 \text{ és } d(v, w) = 0 \iff v = w.$$

$$ii \quad \forall v, w \in V(G) : d(v, w) = d(w, v).$$

$$iii \quad \forall v, w, z \in V(G) : d(v, w) + d(w, z) \geq d(v, z).$$

Bizonyítás. A bizonyítás során négy esetet különböztetünk meg.

1.eset Ha $v = w$, akkor $d(v, v) + d(v, z) = d(v, z)$.

2.eset Ha $w = z$, akkor $d(v, z) + d(z, z) = d(v, z)$.

3.eset Ha $v = z$, akkor $d(z, w) + d(w, z) = 2 \cdot d(w, z) \geq 0 = d(w, z)$.

4.eset Ha v, w, z páronként különböző csúcsok, akkor vegyük a v és w között egy $d(v, w)$ hosszú utat, ugyanígy w és z között egy $d(w, z)$ hosszú utat. Fűzzük össze ezt a két utat, így kapunk v és w között egy $d(v, w) + d(w, z)$ hosszú sétát. Ebből a sétából hagyjuk el az ismétlődő éleket. Ekkor v és z között legfeljebb egy $d(v, w) + d(w, z)$ hosszú utat kapunk. Ekkor a v és z közötti legrövidebb út hossza, a $d(v, z)$, ami legfeljebb $d(v, w) + d(w, z)$ hosszú lehetséges.

□

2.1.10. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. A G gráfot fának nevezzük, ha összefüggő és körmentes. Ha a G gráf csak összefüggő, abban az esetben a gráfot erdőnek nevezzük.

2.1.4. Tétel. Minden fa és minden erdő egyszerű gráf.

Bizonyítás. Legyen $G = (V, E, \varphi)$ véges gráf. Ha G fa, akkor körmentes, így nem tartalmaz egy és két hosszú kört, ezáltal nincs benne sem hurokél, sem párhuzamos él.

□

2.1.5. Tétel. Legyen $G = (V, E, \varphi)$ véges, egyszerű gráf, $|V(G)| \geq 2$. A G gráf, pontosan akkor fa, ha G bármely két különböző csúcsa között pontosan egy út vezet.

Bizonyítás. Először legyen G egy fa. Ekkor G összefüggő, így bármely két különböző csúcsa között legalább egy út vezet. Tegyük fel, hogy $v, w \in V(G)$ különböző csúcsok között legalább két út vezet. Ez a két út valahol szétágazik, majd újra összeér. A két útszakasz egy kört alkot, ami nem lehetséges fa esetén.

Tegyük fel, most G -ben bármely két különböző csúcs között pontosan egy út vezet, így azonnal következik, hogy G összefüggő. Ha G -ben lenne kör, akkor annak két csúcsa között két út is vezetne a kör egyik illetve másik oldalán.

□

2.1.6. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf. Ha G fa, $|V(G)| \geq 2$, akkor G -ben van legalább két elsőfokú csúcs.

Bizonyítás. Vegyük a G gráfban a leghosszabb utat. A körmentesség miatt v_0 nem szomszédos sem $v_2, \dots, v_n$ -nel. Sőt v_0 nem szomszédos az úton bármelyik kívül elhelyezkedő csúccsal sem, mert elé rakva, még hosszabb utat kapnánk, így a v_0 egyetlen szomszédja v_1 , így $d(v_0) = 1$. Hasonlóan alakul, hogy $d(v_n) = 1$. $\square$

2.1.7. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf. Ha G fa, akkor $|V(G)| = |E(G)| + 1$.

Bizonyítás. A tételt $|V(G)|$ szerinti teljes indukcióval látjuk be. $|V(G)| = 1$ esetén $|E(G)| = 0$, így $|V(G)| = |E(G)| + 1$, ami igaz.

Tegyük fel, hogy $|V(G)| \geq 2$ és G -nél kevesebb csúcsú fára igaz az állítás. Ekkor létezik olyan v csúcsa, ami elsőfokú, azaz $\exists v \in V(G) : d(v) = 1$. Ekkor $G - v$ is fa, mert kevesebb csúcsa van, így alkalmazható rá a felvetés.

$$|V(G - v)| = |E(G - v)| + 1 \rightarrow |V(G)| - 1 = (|E(G)| - 1) + 1$$

Azaz $|V(G)| = |E(G)| + 1$. $\square$

2.1.11. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. A G gráf feszítő fája vagy feszítő erdeje olyan feszítő részgráf, ami fa vagy erdő.

2.1.8. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf. A G gráfban pontosan akkor van feszítő fa, ha a gráf összefüggő.

Bizonyítás. Először tegyük fel, hogy G -ben van feszítő fa, akkor a feszítő fában is van bármely csúcsból, bármely csúcsba vezető út, így G -ben is, tehát G összefüggő.

Tegyük fel most, hogy G összefüggő. Ha vannak töröljük a hurokéleket, úgy hogy a gráf még összefüggő maradjon. Ezt addig folytatjuk míg lehet. Így a végső gráf minimálisan összefüggő, tehát fa. Mivel élek törlésével jutottunk ide, ezért feszítő gráf, és így feszítő fa. $\square$

2.1.9. Tétel. Minden gráfban van feszítő erdő.

Bizonyítás. Az előző tétel miatt minden komponensnek van feszítő gráfja, amik így együtt a gráf feszítő erdejét alkotják. $\square$

2.1.12. Definíció. Egy G véges gráfot páros gráfnak nevezünk, ha $\exists A, B \subseteq V(G)$, hogy $A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset, A \cup B = V(G)$ és G minden élének az egyik végpontja A -beli, másik B -beli. Ezeket az A és B osztályokat csúcsosztályoknak nevezzük.

2.1.10. Tétel. Minden legalább két csúcsú erdő páros gráf.

Bizonyítás. Először tegyük fel, hogy az erdők összefüggők, tehát fák. Ezt csúcsok száma szerinti teljes indukcióval lehet bebizonyítani. Ha $|V(G)| = 2$, akkor az egyik csúcs legyen A -beli, másik B -beli. Ekkor ez páros gráf.

Tegyük fel, hogy $|V(G)| \geq 3$ és az ennél kisebb csúcsokra is igaz az állítás. Ekkor G -nek van elsőfokú csúcsa, legyen ez v . Alkalmazzuk az indukciós feltevést $(G-v)$ -re. Ekkor $G-v$ páros gráf, tehát csúcsai jól osztályozhatóak. Ha v szomszédja A -beli, akkor v -t vegyük bele a B csúcsosztályba. Ha v egyetlen szomszédja B -beli, akkor tegyük bele v -t az A csúcsosztályba. Ezzel megadtunk a G fa csúcsainak egy megfelelő osztályozását. Így a G gráf páros gráf.

Tegyük fel, hogy a G erdő nem összefüggő. Ha G üres gráf, akkor kerüljön az egyik csúcs az egyik csúcsosztályba, míg a másik csúcs a másik csúcsosztályba. Ha G nem üres, akkor G -nek van legalább két csúcsú komponense. A legalább két csúcsú komponensekre alkalmazható a bizonyítás előző része, így azok csúcsai megfelelően osztályozhatóak. Ezen osztályozások egyesítésével a legalább két csúcsú komponensek alkotta részgráf csúcsai osztályozva vannak. A kimaradó esetleges izolált csúcsokat tegyük bele valamelyik csúcsosztályba. $\square$

2.1.11. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf, ahol $|V(G)| \leq 2$. A G gráf, pontosan akkor páros gráf, ha benne minden kör páros hosszúságú.

Bizonyítás. Legyen G egy páros gráf, ahol jelölje A és B a csúcsosztályokat. Ekkor egy köre felváltva tartalmaz A és B -beli csúcsokat, így a hossza páros lesz.

Legyen most G egy gráf, ahol $|V(G)| \geq 2$, amiben minden kör páros hosszúságú is. Vegyük a G gráf egy feszítőerdjét. Ami a fenti tétel alapján páros gráf, így a csúcshalmazát megfelelő módon lehet osztályozni. Így az élekre teljesül az, hogy az egyik csúcs az egyik csúcsosztályba, míg a másik csúcs a másik csúcsosztályba található. Legyen ezután $e \in E(G)$, ami nincs a feszítőerdőben. Tekintsük G -nek azt a komponensét, amihez e tartozik. Ennek a komponensnek feszítőgráfja a fenti feszítőerdő egyik komponense. Ez a fa maximálisan körmentes, ezért e behúzásával keletkezik benne kör. Mivel minden kör páros hosszúságú és a többi éle feszítőgráfban van, ezért a kör csúcsai felváltva A és B -beliek, ezért e egyik végpontja A -ban van, míg másik végpontja B -ben van. Tehát G páros gráf. $\square$

2.1.13. Definíció. Egy páros gráfot, A és B csúcsosztályokkal teljes páros gráfnak nevezünk, ha egyszerű és $\forall A$ -beli csúcs szomszédos $\forall B$ -beli csúccsal. Ha $|A| = m$ és $|B| = n$, akkor a teljes páros gráfot $K_{m,n}$ -nel jelöljük.

3. GRACEFUL GRÁFOK ÉS GRACEFUL CÍMKÉZÉS

A bevezető definíciók és tételek után, végre ebben a fejezetben választ kaphatunk, hogy milyen gráfok ezek a bizonyos graceful gráfok, mi a lényege a graceful címkézésnek, illetve hogy milyen alapvető feltételeknek kell teljesülni-e ahhoz, hogy egy gráf graceful legyen. Továbbá ismertetni fogom a graceful gráfok egyik legfontosabb sejtését is.

3.1. Graceful címkézés, illetve címkézés teljes gráfok és fák esetén

A graceful címkézés Solomon W. Golomb nevéhez fűződik, de elsőként Alexander Rosa cikkében olvashatunk a graceful gráfok problémáiról, de Rosa cikkében egy másik nevet használ ezekre a címkézésekre, mégpedig a β -címkézést.

3.1.1. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. Ekkor a G gráf csúcsainak a címkézésén, egy olyan bijektív Ψ leképezést értünk, ahol a $V(G)$ elemeit, nem negatív egész számokkal feleltetjük meg. Hasonlóan beszélhetünk így élek címkézéseiről is.

3.1.2. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = n$, $|E(G)| = m$, $v, w \in V(G)$ és az v, w csúcsokat a $e \in E(G)$ él köti össze. A G gráfnak létezik graceful címkézése, ha van olyan $V(G) \rightarrow \{0, \dots, m\}$ injektív függvény, hogy $\forall e$:

$$|f(v) - f(w)|$$

különböző. Ekkor $E(G) = \{1, 2, \dots, m\}$.

3.1.1. Lemma. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = m$ és $|E(G)| = n$. Ha a G gráf graceful, akkor $m \leq n + 1$.

3.1.1. Tétel. Legyen $G = (V, E, \varphi)$ véges, teljes gráf és $|V(G)| = n$. Ha $n = 2, 3, 4$, akkor a G gráfnak van graceful címkézése. Ha $n > 4$, akkor a G gráfnak nincs graceful címkézése.

Bizonyítás. Először legyen $|E(G)| = m$, ami teljes gráf esetén $m = \binom{n}{2} \geq 10$. Indirekt, tegyük fel, hogy a G gráfnak van graceful címkézése. Legyen $V(G) \subseteq \{0, 1, \dots, m\}$, továbbá legyen az éleknek a címkéi különböző egész számok 1-től m -ig.

Jelöljük a G gráfnak a 0 és az m csúcs címkéit összekötő élet m -mel ($|m - 0| = m$). Annak érdekében, hogy egy él címkéje $m - 1$ legyen, vagy 1-nek vagy $m - 1$ -nek kell lenni-e a csúcs címkézésének ($|(m - 1) - 0| = m - 1$ vagy $|m - 1| = m - 1$). Így bármely G graceful gráf esetén, amelynek m éle van, úgy minden a_i csúcs címkéje kicserélhető $m - a_i$ -val, úgy hogy az élek címkézése nem változik. Így az általánosság megszorítása nélkül meg tudjuk változtatni az 1 csúcs címkéjét az $m - 1$ csúcs címkéjével.

Jelölje a G gráfnak az egyik él címkéjét $m - 2$. Ez az él címke pedig, az $m - 1$ és az m csúscímkézett csúcsokat összekötő élnek a címkéje lesz. ($|(m - 1) - 1| = m - 2$). Ha két él címkézés is 1 lenne, akkor az egyik a 0 és az 1 csúcsok között helyezkedne el, míg a másik az m és az $m - 1$ csúcsok között. Ha egy csúcs címkéje 2 lenne, akkor az $m - 2$ él kötné össze őt az m csúcs címkével jelölt csúccsal ($|m - 2| = m - 2$), de újra lenne 2 olyan él címke, aminek a címkéje 1 lenne, az egyik a 0 és az 1 csúcs között lenne, a másik az 1 és a 2 csúcs között szerepelne. Ekkor a csúcsok címkéi 0, 1, $m - 2$ és m , míg az élek címkéi 1, 2, $m - 3$, $m - 2$, $m - 1$ és m . Annak érdekében, hogy egy él címkéje $m - 4$ legyen, kell hogy legyen egy negyedik csúcs címke. Legyen ezért a következő csúcs címkéje 2. Ekkor azonban lenne két olyan él címke, aminek a távolsága 2 lenne, ez a két él pedig a 2 és a 0, illetve az m és az $m - 2$ él közötti élek lennének. Ha $n = 5$, akkor következőt kell tenni. (Pontosan 5 csúcs címkénk van). Azonban, ebben az esetben $m = 10$, és így lesz $m - 6 = 4$ él címke, így van olyan él címkézés, ahol két különböző él címkéje megegyezik. Így $n = 5$ esetén nincs jó címkézés. Most vizsgáljuk meg azokat az eseteket, ahol $n > 5$. Ebben az esetben $m > 10$ és $4 < m - 6$, így ilyenkor nem lesz él ismétlődés. Ekkor azonban a következő csúcs címkéink lesznek: 0, 1, 4, $m - 2$, m . Illetve a következő él címkéink lesznek: 1, 2, 3, 4, $m - 6$, $m - 4$, $m - 3$, $m - 2$, $m - 1$, m .

Azonban nincs olyan módszer, ahol az él címkéje $m - 5$ legyen, mert minden választás szerint, kapunk olyan élcímkézést, amely legalább egy csúscímkézést tartalmaz, amit viszont így már nem lehet használni. Adjuk hozzá az $m - 5$ csúcs címkét és az $m - 6$ él címkét. Továbbá adjuk hozzá az $m - 4$ csúcs címkét, amiből az él az $m - (m - 4)$ -ből adódik, azaz 4 lesz. Így az $m - 1$ csúcs címke esetén, az él az $(m - 1) - 1$ -ből adódik, ami $m - 2$ lesz. Nem lehet már hozzárendelni a 3 csúcs címkét, amiből az él a $3 - 1 = 2$ adódna. Végül rendeljük az utolsó csúcshoz az 5 csúcs címkét. Ez viszont nem lehetséges, mivel két olyan élt kaptunk, aminek a felirata 1 lesz. Így egy ellentmondást kaptunk, így $n \geq 5$ esetén a G gráf már nem lesz graceful. $\square$

3.1.2. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf, $|E(G)| = m$. Ha a G gráfban van zárt Euler-vonal és $m \equiv 1 \pmod{4}$ vagy $m \equiv 2 \pmod{4}$, akkor a G gráf nem graceful gráf.

Bizonyítás. Tegyük, fel indirekt, hogy a G gráf mégis graceful, úgy hogy tartalmaz zárt Euler-vonalat és $m \equiv 1 \pmod{4}$ vagy $m \equiv 2 \pmod{4}$. Összegezzünk minden e_i él címkeére. Ekkor $\sum_{i=1}^m e_i = \sum_{i=1}^m i$, ami egyenlő $\frac{(m+1) \cdot m}{2}$. Ha $m \equiv 1$ vagy $m \equiv 2 \pmod{2}$, akkor $\sum_{i=1}^m e_i \equiv 1 \pmod{2}$. Azonban, egy tetszőlegesen körön, mondjuk $v_1, v_2, \dots, v_n, v_1$, az élek címkézése $|v_1 - v_2|, \dots, |v_m - v_1|$, és mivel y abszolút értéke $\pmod{2}$ -ben kongruens y -nal, ezért az abszolút értéket elhagyhatjuk, így $\sum_{i=1}^{m-1} |v_i - v_{i+1}| + |v_m - v_1|$ kapjuk. Amit, ha tovább számolunk $\sum_{i=1}^{m-1} (v_i - v_{i+1}) + v_m - v_1 \equiv 0 \pmod{2}$ kapjuk. Ami ellentmondás, mert $\sum_{i=1}^m e_i \equiv 1 \pmod{2}$, ami állításunkat igazolja. $\square$

3.1.3. Tétel. Legyen $G = (V, E, \varphi)$ véges gráf. Ha a G gráf út gráf, akkor a G gráf graceful.

Bizonyítás. A bizonyítás konstrukcióval történik, azaz egy címkézést adunk, hogyan lehet egy G gráfot, ami út gráf, elcímkézni úgy, hogy graceful legyen. Legelőször is a gráf legelső csúcsának a címkéje legyen 0 és minden más páratlan csúcsnak a címkéjét 1-gyel növeljük. A második csúcsnak a címkéje legyen s és minden más páros csúcsnak a címkéből vonjunk ki 1-et. Így $s + 1$ csúcsunk van, és az $\{0, 1, \dots, s + 1\}$ halmazból felcímkézzük a csúcsokat. Így az első esetben $\{0, 1, \dots, \frac{s}{2}\}$, míg a második esetben $\{\frac{(s+2)}{2}, \dots, s\}$ lesz a két halmaz, amiből a csúcsokat felcímkézzük. Ez a konstrukció helyes, így az állításunkat beláttuk. $\square$

Amikor Alexander Rosa el kezdte vizsgálni a gráfoknak a graceful címkézéseit, felvetődött az a kérdés, hogy vajon minden gráf graceful-e, vagy vannak-e a gráfoknak egy olyan osztálya, ami biztosan az. Ringel és Kotzig is Rosa után el kezdett vizsgálgódn a graceful "jelenséggel" és kialakult egy sejtés bennük.

3.1.1. *Sejtés* (Ringel-Kotzig sejtés). Minden fának létezik graceful címkézése.

Sok ember próbálkozott ezt a sejtést belátni, de ez az 50 éves probléma még a mai napig is csak egy sejtés maradt. Azonban Rosa és társai, ezen sejtés belátásához el kezdték a fa gráfokat még jobban elemezni. Így elkezdtek a fákat bizonyos tulajdonságok alapján külön osztályokba sorolni, így beszélhetünk ma már csillag gráfokról, caterpillar gráfokról, super-caterpillar gráfokról, lobster gráfokról, spider gráfokról.

3.1.3. Definíció. Egy gráf levelén, olyan csúcsot értünk, amelynek a fokszáma pontosan 1.

3.1.4. Definíció. Az út gráf, olyan speciális fa gráf, amelynek pontosan két csúcsa levél.

3.1.5. Definíció. A caterpillar fa gráf, olyan speciális fa gráf, amely egy központi útból áll és tetszőleges számú levél csatlakozik az úthoz.

3.1.1. *Megjegyzés.* A caterpillar gráf definíciójából, azonnal következik, hogy $|V(G)| \geq 2$. Ha $|V(G)| \geq 3$, akkor a caterpillar gráfnak létezik központi útja. Illetve, ha bizonyos leveleket eltávolítunk a gráfból, akkor egy út gráfot kapunk.

A most következő tétel, a caterpillar gráfok graceful tulajdonságáról szól, amit a későbbiek folyamán felhasználok a graceful caterpillar gráf megalkotásához.

3.1.4. Tétel. *Minden caterpillar fa gráfnak van graceful címkézése.*

Bizonyítás. Legyen G egy caterpillar gráf, ahol $|V(G)| = n$. Ha $n \leq 2$, akkor G útjának a hossza 1, ami caterpillar gráf. Így feltehetjük, hogy a csúcsok száma nagyobb vagy egyenlő háromnál, és továbbá feltesszük, hogyha a G gráfból leveleket törölünk, akkor az U utat kapjuk meg, ami $v_0, v_1, \dots, v_k$. Mivel $n \geq 3$, ezért az U utunk legalább egy csúccsal rendelkezik. Továbbá válasszuk szét a csúcsokat az alábbiak szerint:

$$X = \{x \mid x \in V(G), d(v_0, x) \equiv 0 \pmod{2}\}$$

$$Y = \{y \mid y \in V(G), d(v_0, y) \equiv 1 \pmod{2}\} = V(G) - X.$$

Ez azt jelenti, hogy $v_i \in X$ ha i páros, és $v_i \in Y$, ha i páratlan. Legyen v_0 címkéje n , v_0 szomszédainak címkéje $1, 2, 3, \dots$, ahol a legnagyobb címkével ellátott csúcs legyen v_1 . Rendeljük hozzá az $n-1, n-2, n-3$ címkéket a v_1 szomszédjaihoz, kivéve v_0 -hoz, és a legnagyobb címke legyen v_2 . Miután v_{2i} is megkapja a címkéjét, rendeljünk növekvő számokat a szomszédjaihoz, kivéve a v_{2i-1} -hez, úgy hogy a legkisebb nem használt címkével címkézzünk. Végül a legnagyobb megmaradt címkét hozzárendeljük a v_{2i+1} csúcsokhoz. Ezután a szomszédait címkézzük úgy a v_{2i+1} -nek (kivéve a v_{2i} csúcsokat), hogy a legnagyobb számú megmaradó címkét csökkenő sorrendben kiosszuk a csúcsoknak. Végül a legkisebb megmaradt címkét hozzárendeljük a v_{2i+2} csúcsokhoz. Ezen az alapon történő címkézés során, az összes címkénk elfogyott az X és az Y halmazokból, így az összes csúcsnak van címkézése, ráadásul a csúcsok címkézése különböző lesz. Legyen l_i a v_i csúcs címkéje, és így $l_1 < l_3 < \dots$ és $l_0 > l_2 > \dots$. Legyen i páros, és ha $i \notin \{0, k\}$ akkor a szomszédai a v_i csúcsának a címkézése legyen $l_{i-1}, l_{i-1}+1, \dots, l_{i+1}$ és az indukált él címkék így $l_i - l_{i+1}, l_i - l_{i+1} - 1, \dots, l_{i+1}$ lesznek. Ha i páratlan, akkor $i \neq k$, így a v_i szomszédainak a címkéi $l_{i-1}, l_{i-1}+1, \dots, l_{i+1}$ lesznek és az indukált él címkék pedig $l_{i-1} - l_i, l_{i-1} + 1 - l_i, \dots, l_i + 1 - l_i$ lesznek. v_0 szomszédainak a címkézése $1, 2, \dots, l_1$ lesz, míg v_k szomszédainak címkéi $l_{k-1}, l_{k-1}+1, \dots, l_k-1$ lesznek, ha k páros, és $l_k+1, l_k+2, \dots, l_{k+1}$, ha k páratlan. Könnyen látható ebből, hogy az $n-1$ él címke mind különböző lesz, és a legnagyobb $n-1$, míg a legkisebb 1 lesz. Így ez egy alkalmas graceful címkézés. $\square$

3.1.6. Definíció. Legyen adva egy caterpillar gráf. Ha a caterpillar gráf bármelyik csúcsához több levél is csatlakozik, akkor a gráfunkat lobster gráfnak nevezük.

Ezekről a gráfokról érdemes tudni, hogy 1979-ben Bermond azt sejtette, hogy minden lobster gráf graceful. Azonban nem tudta bebizonyítani, és ez csak egy sejtés maradt. 2008-ban Mishra és Panigrahi be tudták látni, hogy minden olyan lobster gráf, amelyiknek az átmérője legalább öt az már graceful [7]. Illetve egy másik osztályáról is be tudták látni, hogy az is graceful. Tehát a lobster gráfoknak a graceful címkézése, még mindig nyitott kérdés.

3.1.7. Definíció. Egy fa gráfot, szimmetrikus fának nevezünk, ha a fa minden szintjén ugyanannyi csúcsot találunk.

3.1.5. Tétel. *Minden szimmetrikus fának van graceful címkézése.*

3.1.2. *Megjegyzés.* A tétel bizonyítását a [7] cikkben olvashatjuk, melyet teljes indukcióval látnak be.

3.1.8. Definíció. Egy fa gráfot, csillag gráfnak nevezünk, ha $k + 1$ csúcsa van és a $k + 1$ csúcsból, pontosan k db levél van.

3.1.3. *Megjegyzés.* Az utak és a csillagok is egyszerű caterpillar gráfok. Az út esetén a caterpillar gráfnak pontosan két levele van, míg a csillagnál legalább két levél mindig van, és a központi útvonala hossza 0.

3.1.9. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. Ekkor a G gráf átmérőjén, a leghosszabb út hosszát értjük, amit $\text{diam}(G)$ -vel szokás jelölni.

3.1.6. Tétel. *Legyen $G = (V, E, \varphi)$ véges fa gráf, illetve $\text{diam}(G) = d$ és $1 \leq d \leq 3$. Ekkor a G fa gráf, caterpillar fa gráf.*

Bizonyítás. Tegyük fel, hogy a G gráfnak az átmérője i , ahol $i \in \{1, 2, 3\}$. Ha $i = 1$, akkor egyértelmű, hogy a G gráf a két csúcsú teljes gráf.

Ha $i = 2$, akkor legyen $P = v_0v_1v_2$ a leghosszabb út a G gráfban. Ekkor azonban v_0 és v_2 levelek, különben az út hosszabb lenne. Így, ha y egy másik csúcs lenne, akkor az y és a v_1 csúcs között is lenne út, viszont az nem megy át a v_0 és v_1 csúcsokon. Viszont v_0 hozzáadásával tudjuk növelni az út hosszát. Mivel két hosszúságú a leghosszabb út, ez viszont azt jelenti, hogy y -t a v_1 -hez kell hozzácsatolni. Így G egy csillag lesz, ami caterpillar gráf, tehát készen vagyunk. Végül, ha $i = 3$, akkor legyen $P = v_0v_1v_2v_3$ a leghosszabb út G -ben. Az megint világos, hogy v_0 és v_3 lesznek a levelek, mert akkor a P utat lehetne növelni. Ha létezik egy y csúcs G -ben, akkor létezni-e kell egy y -ból induló útnak, ami nem tartalmaz más csúcsot a P útból. Ez azt jelenti, hogy létezni kell egy olyan útnak ami y -ból indul és v_1 vagy v_2 -ben ér véget. Tegyük fel, hogy az út v_1 -ben

végződik, de v_2 -re is ugyanez a bizonyítás. Ha az út y -ból v_1 -ben nem csak egy éllel csatlakozik, akkor létezni-e kell egy olyan útnak, amelynek hossza legalább négy, a v_2 és v_3 csúcsokkal együtt. Így bármely olyan y csúcs esetén, ami nincs rajta a P úton, az csak egy éllel csatlakozhat a v_1 -hez vagy a v_2 -höz. Így a gráfunk csak caterpillar gráf lehet. $\square$

3.1.7. Tétel. Minden $G = (V, E, \varphi)$ véges fa gráf, ahol $\text{diam}(G) = 4$ teljesül, az graceful gráf.

3.1.8. Tétel. Minden $G = (V, E, \varphi)$ véges fa gráf, ahol $\text{diam}(G) = 5$ teljesül, az graceful gráf.

3.1.4. *Megjegyzés.* A két tétel bizonyítását a [7] cikkben olvashatjuk, melyet a cikkben található, 20-as lemma segítségével bizonyítottak.

3.1.10. Definíció. Egy G gráfot spider fa gráfnak nevezünk, ha fa gráf és pontosan egy olyan csúcsa van, amelyiknek fokszáma nagyobb, mint 2. Ha létezik, ilyen csúcs, akkor ezt a csúcsot elágazási pontnak nevezzük, illetve a spider gráfban az elágazási pontból induló különböző hosszúságú utakat nevezzük a spider gráf lábainak.

3.1.9. Tétel. Legyen $G = (V, E, \varphi)$ véges spider gráf s darab lábbal, amelyeknek hossza $\{n, n+1\}$, ahol $n > 1$. Ekkor a G gráfnak létezik graceful címkézése.

3.1.5. *Megjegyzés.* A tétel bizonyítása a [7] cikkben olvasható, mely egy konstrukatív bizonyítás.

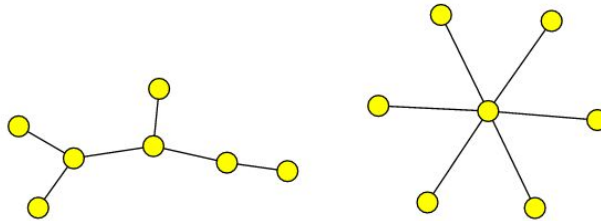
3.1.10. Tétel. Egy spider gráf, ami 3 útból vagy 4 útból áll, az graceful.

3.1.6. *Megjegyzés.* A tétel bizonyítását a [5] tanulmányában olvashatjuk, mely a 3.16-os lemmát használja fel a bizonyítás során.

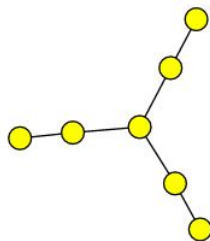
3.1.11. Tétel. Legyen $G = (V, E, \varphi)$ véges spider gráf p útból. Ha az utak mindegyike, k hosszúságú, ahol $k \in \mathbb{N}$, akkor a spider gráf graceful.

3.1.7. *Megjegyzés.* A tétel bizonyítását a [5] tanulmányban olvashatjuk.

3.1. ábra. Caterpillar gráf és csillag gráf



3.2. ábra. Spider gráf



3.1.11. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf. A G gráfot firecracker gráfnak nevezzük, ha egy tetszőleges n hosszú útból áll és az úton lévő összes csúcs, egy éllel van összekötve, pontosan egy különböző csillag gráfhoz.

3.1.12. Tétel. *Minden firecracker gráfnak van graceful címkézése.*

3.1.8. *Megjegyzés.* A bizonyítást szintén a [7] cikkben olvashatjuk.

Végül a legfontosabb tétel és egyben ennek a résznek a záró tétele megmutatja, hogy a fa gráfokon belül is, mennyi az a maximális csúcs, amiről már be tudták látni, hogy a gráfnak létezik graceful címkézése.

3.1.13. Tétel. *Minden fa amelynek legfeljebb 27 csúcsa van, annak létezik graceful címkézése.*

Ezt a tételt Aldred és McKay bizonyította be számítógép használataival, melynek menetét a [3] és [5] tanulmányokban olvashatjuk.

4. KÜLÖNLEGES GRÁFOK, KONSTRUKCIÓK ÉS SAGE IMPLEMENTÁCIÓ

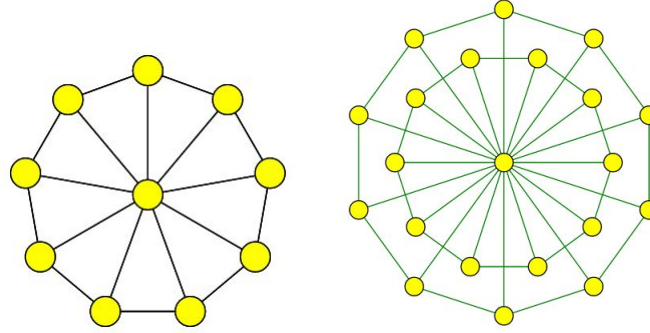
Ebben a fejezetben szeretnék megmutatni pár eléggé különleges gráfot, amiket a nevük alapján is el tudunk már képzelni. Ezek a gráfok mind gracefullak, így egy program segítségével meg lehet konstruálni ezeket az egyszerű gráfokat. Ehhez segítségül a Sage programcsomagot használom, amelyhez könnyen hozzá lehet jutni az interneten, és a programozás nem annyira bonyolult, mint más programnyelvek esetén. A Sage a Python nyelvet használja, amely egy magas szintű programozási nyelv, de a legnagyobb előnye mégis az, hogy a nyelv folyamatosan fejlődik.

4.1. Graceful double-wheel gráf konstrukciója

4.1.1. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = n$. Wheel gráfnak nevezünk egy gráfot, ha a központi csúcsa minden csúccsal össze van kötve, a többi csúcs viszont csak a központi csúcshoz vezető él kivételével, pontosan két csúccsal van összekötve. Jele: W_n .

4.1.2. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = 2 \cdot N + 1$, ahol N azon csúcsok száma, amik egyes körökön helyezkednek el. Double-wheel gráfról beszélünk, hogyha a körön lévő összes csúcs csatlakozik a központi csúcshoz, ahol a központi csúcs egyik körön sem helyezkedik el. Jele: DW_N .

4.1. ábra. Wheel gráf (W_{10}) és double-wheel gráf(DW_{10})



Vegyük elsőként a double-wheel gráfokat, amiket az alábbi feltételek szerint tudunk megkonstruálni.

Legyen N a körökön lévő csúcsok száma és x_i az a csúcs, ahol az i -edik csúcs címkéje ezen a belső körön van. Továbbá legyen y_j az a csúcs, ahol a j -edik csúcs címkéje a második (külső) körön van, végül z legyen a központi csúcs egy címkéje. A javasolt konstrukció során két esetet különböztetünk meg, attól függően, hogy milyen az N paritása. Mindkét esetben legyen a központi csúcs (z) címkéje 0. Továbbá a csúcsok címkézései a két körön az alábbiak szerint legyenek definiálva:

Ha N páros és $N \geq 4$:

$$x_i \begin{cases} i & , \text{ ha } i \text{ páratlan és } 1 \leq i \leq N-3 \\ 4 \cdot N - 3 - i & , \text{ ha } i \text{ páros és } 2 \leq i \leq N-2 \\ 4 \cdot N & , \text{ ha } i = N-1 \\ 4 \cdot N - 2 & , \text{ ha } i = N. \end{cases}$$

$$y_j \begin{cases} 4 \cdot N - 4 & , \text{ ha } j = 1 \\ N + j & , \text{ ha } j \text{ páratlan és } 3 \leq j \leq N-3 \\ 3 \cdot N - 1 - j & , \text{ ha } j \text{ páros és } 2 \leq j \leq N-2 \\ 4 \cdot N - 1 & , \text{ ha } j = N-1 \\ 2 \cdot N & , \text{ ha } j = N. \end{cases}$$

Ha N páratlan és $N \geq 5$:

$$x_i \begin{cases} i & , \text{ha } i \text{ páratlan és } 1 \leq i \leq N-2 \\ 4 \cdot N + 1 - i & , \text{ha } i \text{ páros és } 2 \leq i \leq N-3 \\ 2 \cdot N & , \text{ha } i = N-1 \\ 2 \cdot N + 4 & , \text{ha } i = N. \end{cases}$$

$$y_j \begin{cases} 4 \cdot N & , \text{ha } j = 1 \\ 2 \cdot N + 2 + j & , \text{ha } j \text{ páratlan és } 3 \leq j \leq N-2 \\ 2 \cdot N + 1 - j & , \text{ha } j \text{ páros és } 2 \leq j \leq N-3 \\ 3 \cdot N + 2 & , \text{ha } j = N-1 \\ 2 \cdot N + 2 & , \text{ha } j = N. \end{cases}$$

Az itt látható programkód, az előző feltételeknek az implementálása, ami egyszerűen pár darab `for` ciklussal, és pár darab `if` paranccsal készült. Így kapunk egy tömböt, melyben szerepelni fognak a graceful címkézés során felhasznált csúcscímkék.

```
def Labels(n):
    L=[]
    if n % 2 == 0 and 4<=n :
        for i in range(1,n+1):
            if i==n:
                L.insert(len(L),4*n-2)
            elif i==n-1:
                L.insert(len(L),4*n)
            elif i % 2 <>0 and 1<=i and i<=n-3:
                L.insert(len(L),i)
            elif i % 2 ==0 and 2<=i and i<=n-2:
                L.insert(len(L),4*n-3-i)
        for j in range(1,n+1):
            if j==1:
                L.insert(len(L),4*n-4)
            elif j % 2 <>0 and 3<=j and j<=n-3:
                L.insert(len(L),n+j)
            elif j % 2 ==0 and 2<=j and j<=n-2:
                L.insert(len(L),3*n-1-j)
            elif j==n-1 :
                L.insert(len(L),4*n-1)
            elif j==n:
                L.insert(len(L),2*n)
    elif n % 2 <> 0 and n>=5:
```

```

for k in range(1,n+1):
    if k==n-1:
        L.insert(len(L),2*n)
    elif k==n:
        L.insert(len(L),2*n+4)
    elif k % 2 <>0 and 1<=k and k<=n-2:
        L.insert(len(L),k)
    elif k % 2 == 0 and 2<=k and k<=n-3:
        L.insert(len(L),4*n+1-k)
for l in range(1,n+1):
    if l==1:
        L.insert(len(L),4*n)
    elif l % 2 <>0 and 3<=l and l<=n-2:
        L.insert(len(L),2*n+2+l)
    elif l % 2 ==0 and 2<=l and l<=n-3:
        L.insert(len(L),2*n+1-l)
    elif l==n-1:
        L.insert(len(L),3*n+2)
    elif l==n:
        L.insert(len(L),2*n+2)
return L

```

A double-wheel gráf szomszédsági mátrixát használjuk fel a gráf konstrukciójához, mert a mátrixokat könnyű kezelni. Ami egy 9 csúcsú gráfnál a következőképpen néz ki, de a programunkban is egy hasonló mátrix van leprogramozva, csak a program bármilyen n -re képes ezt megkonstruálni.

A double-wheel gráf szomszédsági mátrixa a következő képpen nézz ki:

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

```
def adjmatrix_row(n,i):
    row=[]
    I1 = I2 = 0
    if i < 2:
        I1 = 2 + i
        I2 = 2 * n - (2 - i)
    elif i > (2 * n - 3):
        I1 = i - (2 * n - 2)
        I2 = 2 * n - 4 + I1
    else:
        I1 = i - 2
        I2 = I1 + 4
    for j in range(2*n):
        if j == I1 or j == I2:
            row.insert(j, 1)
        else:
            row.insert(j, 0)
    row.insert(len(row), 1)
    return row
def dwgraf_adjmatrix(a):
    M=Matrix(2 * a + 1)
    if a > 2:
        for i in range(2 * a):
            aktRow=adjmatrix_row(a, i)
            M.set_row(i, aktRow)
    lastRow = []
```

```

for i in range(a * 2):
    lastRow.insert(len(lastRow), 1)
    lastRow.insert(len(lastRow), 0)
    M.set_row(M.nrows()-1, lastRow)
return M

```

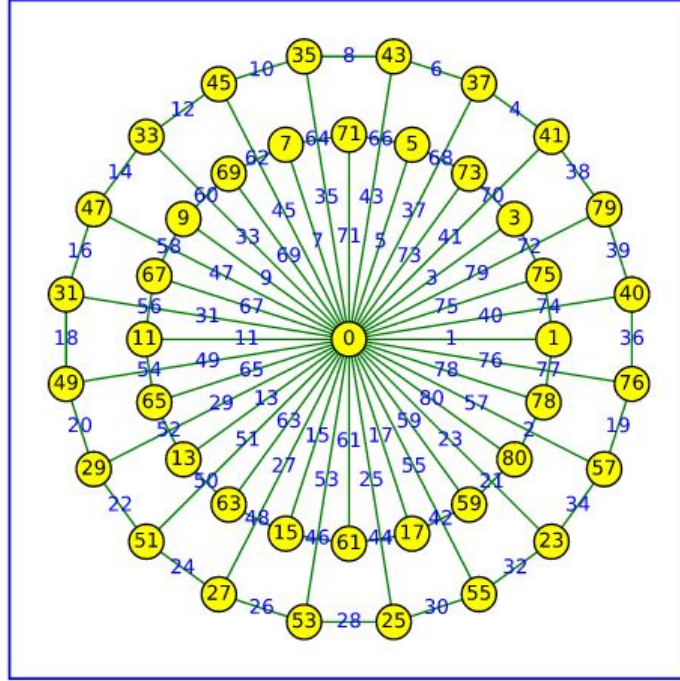
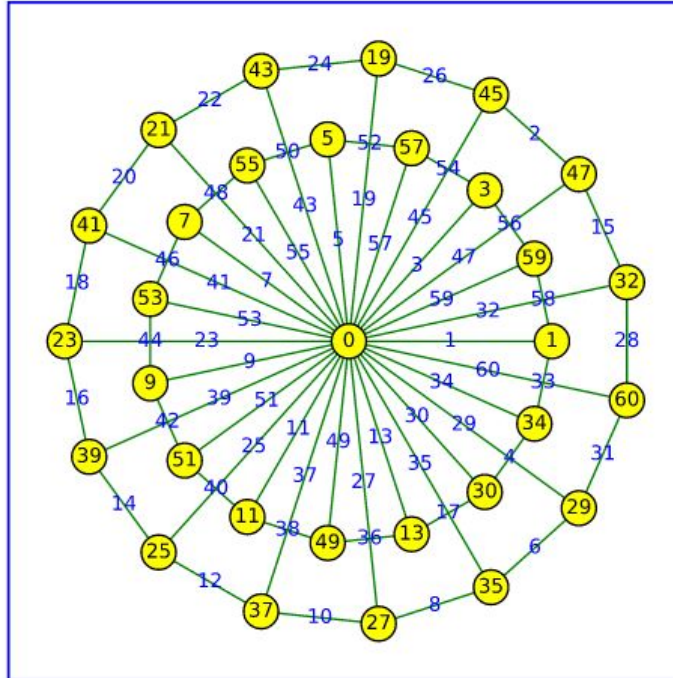
Végül már csak a gráf megrajzoltatása és a címkézése maradt hátra. A megrajzoltatása a `draw` függvénnyel történik. Ebben a függvényben a szomszédsági mátrix alapján megkonstruáljuk a gráfot, viszont ügyelni kell a csúcsok elhelyezkedésére. Ha nem pozicionáljuk a csúcsokat, akkor igaz, hogy a szomszédsági mátrix alapján minden csúcs jó csúccsal van összekötve, de nem kör alakú gráfokat kapunk. A csúcsok pozíciójának a megadása a következőképpen zajlik. A nulla csúscímkevel ellátott csúcs, mindig a központi csúcs legyen, aminek a helyét x és y koordinátákkal adjuk meg. Így a nullás címke a $[0, 0]$ lesz, azaz ő lesz középen. A további csúcsok x koordinátáját $(6 - (-1)^i) \cdot \text{float}(\cos((\frac{2 \cdot \pi}{2 \cdot n}) \cdot i))$ képlettel számolhatjuk ki. Az y -t is hasonlóképpen tudjuk számolni, csak abban az esetben a koszinusz függvény helyett a szinusz függvényt használunk. Már csak a címkézés van hátra, amit a definíció alapján írunk be, majd a feltételeket hozzárendeljük, és megkapjuk az eljárást a graceful double-wheel gráfok megkonstruálásához.

```

def DW(n):
    R=Labels(n)
    T1=[]
    T2=[]
    for k in range(len(R)/2):
        T1.insert(len(T1),R[k])
        T2.insert(len(T2),R[len(R)-1-k])
    draw(n, T1, T2)
def draw(n, T1, T2):
    F=Graph(dwgraf_adjmatrix(n))
    T=[]
    for i in range(len(T1)):
        T.insert(len(T),T1[i])
        T.insert(len(T),T2[i])
    T.insert(len(T),0)
    from math import sin, cos, pi
    pos_dict = {}
    for i in range(len(T)-1):
        x = (6-(-1)^i)*float(cos(((2*pi)/(2*n))*i))
        y = (6-(-1)^i)*float(sin(((2*pi)/(2*n))*i))
        pos_dict[T[i]] = [x,y]

```

```
pos_dict[T[len(T)-1]]= [0,0]
F.relabel( [ T[i] for i in range(F.order()) ],
inplace=True)
for u,v,l in F.edges(): F.set_edge_label(u,v,abs(u-v))
pl = F.graphplot(pos=pos_dict,edge_labels=True,
edge_color='green',vertex_colors='yellow',
vertex_size=300,graph_border=True)
pl.show(figsize=8)
html('<h3>Double-Wheel Graceful graph</h3>')
@interact
def _(n=slider(4,25,1,1,
'Adja meg a gráf köreinek a csúcsszámát!',True)):
    [DW(n)]
```

4.2. ábra. Páros double-wheel gráf címkézéssel (DW_{20})4.3. ábra. Páratlan double-wheel gráf címkézéssel (DW_{15})

A double-wheel gráfok esetén az algoritmus még barátságosnak is mondható, de felvetődhet az a kérdés, hogy mi van, ha nem csak két körünk van, hanem három, azaz triple-wheel gráf esetén. A triple-wheel gráfok esetén szeretnék egy másik programot is felhasználni, mégpedig a MiniZinc programot. A program egyedi nyelvvel bír, egyszerűbb például, mint a Sage programcsomag. A programmal főleg különböző optimalizálási feladatokat tudunk elvégezni, de segítségünkre van, ha szeretnénk triple-wheel gráfokat vizsgálni.

Elsőként írni kell egy programot, ami a definíción alapul, tehát a csúcsokat megszámozzuk és ,amelyik csúcsból él vezet ki egy másik csúcsba, azoknak a csúcsok címkéjének az abszolút különbségét vesszük. A k_1 , k_2 , k_3 az inputok, ahol meg kell adni, hogy egy körön hány darab csúcs helyezkedjen el. A mi esetünkben most legyen mindegyik körön három-három csúcs. Az m érték jelenti az élek számát, az n pedig jelentse a csúcsok számát. Ami a MiniZinc nyelvén a következő:

```
include "globals.mzn";

int: k1 = 3;
int: k2 = 3;
int: k3 = 3;
int: m = 2*k1+2*k2+2*k3;
int: n = k1+k2+k3+1;
array[1..n] of var 0..m: nodes;
array[1..m] of var 1..m: edges;

solve :: int_search(nodes, first_fail, indomain_min, complete) satisfy;

constraint
  forall(a in 1..k1+k2+k3)(
    abs(nodes[1]-nodes[a+1]) = edges[a]
  )
  /\
  abs(nodes[2]-nodes[k1+1]) = edges[k1+k2+k3+1]
  /\
  forall(a in k1+k2+k3+2..2*k1+2*k2+k3)(
    abs(nodes[a-k1-k2-k3]-nodes[a-k1-k2-k3+1]) = edges[a]
  )
  /\
  abs(nodes[k1+2]-nodes[k1+k2+1]) = edges[2*k1+2*k2+1]
  /\

  abs(nodes[k1+k2+2]-nodes[k1+k2+k3+1]) = edges[2*k1+2*k2+k3+1]
  /\
```

```
forall(a in 2*k1+2*k2+k3+2..2*k1+2*k2+2*k3)(
    abs(nodes[a-k1-k2-k3]-nodes[a-k1-k2-k3+1]) = edges[a]
)
/\
all_different(edges)
/\
all_different(nodes)
;

output
[
    "nodes: " ++ show(nodes) ++ "\n"
];
```

A válasz pedig a következő:

```
Compiling triple-wheel2.mzn
Running triple-wheel2.mzn
=====UNSATISFIABLE=====
Finished in 6s 10msec
```

Ez az üzenet azt jelenti, hogy az általunk készített program nem talált, olyan számsorozatot, amelynél minden él különböző lenne. Mivel a TW_3 gráf esetén, nem követeltünk meg semmilyen feltételt sem, csak a definíciót használtuk, kijelenthető, hogy a TW_3 esetén nincs jó graceful címkézés.

Most vizsgáljuk meg, azt az esetet, amikor minden körön négy darab csúcs található. Ebben az esetben viszont létezik megoldás. Sőt ha azt is megköveteljük, hogy a központi csúcs címkéje legyen 0, és a $k1$ körön legyen az 1-es csúscímke, akkor is van megoldás.

Körök	1. csúcs	2. csúcs	3. csúcs	4. csúcs
1. kör	1	20	23	9
2. kör	15	5	17	21
3. kör	11	18	2	24

Körök	1. csúcs	2. csúcs	3. csúcs	4. csúcs
1. kör	1	18	13	24
2. kör	15	12	16	6
3. kör	14	22	2	21

Ez a két táblázat azt mutatja meg, hogy a körökön mely csúscímkék helyezkedik el. A 0 címke azért nem szerepel a táblázatban, mert az a központi csúcs. Ez csak két megoldás a TW_4 gráfra, de ezeken kívül is számos megoldás létezik.

Hasonló a helyzet, hogyha olyan gráfot nézünk, amelyiknek bár két kereke van, de a kerekeken a csúcsok száma nem egyezik meg. Ebben az esetben csak két körünk lesz, k_1 és k_2 , az n és az m ugyanúgy az élek, illetve a csúcsok számát jelöli. Ha megnézzük a $2 - 2$, $2 - 3$, $3 - 3$ eseteket, akkor azt tapasztaljuk, hogy nincs graceful címkézés, mert a UNSATISFIABLE feliratot kaptuk vissza. Viszont megvizsgálva a $3 - 4$ esetet már kapunk megoldásokat.

```

Compiling dw.mzn
Running dw.mzn
csucsk: [0, 1, 6, 13, 10, 2, 11, 14]
-----
csucsk: [0, 1, 6, 13, 11, 2, 10, 14]
-----
csucsk: [0, 1, 13, 6, 10, 2, 11, 14]
-----
csucsk: [0, 1, 13, 6, 11, 2, 10, 14]
-----
=====
Finished in 99msec

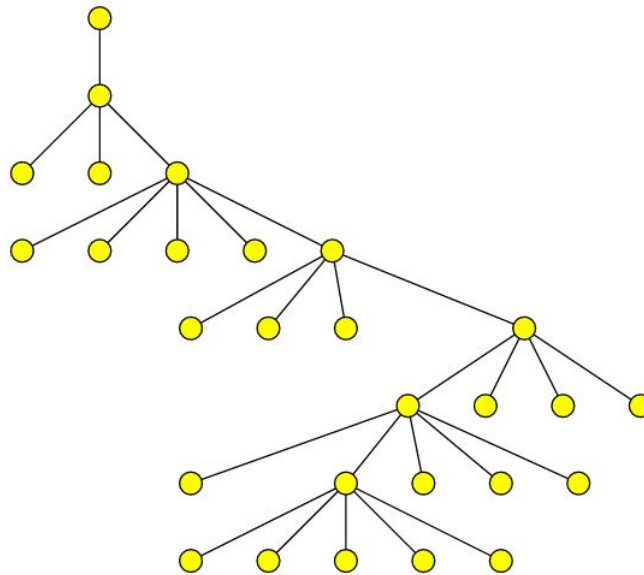
```

A program futtatása előtt, megköveteljük, hogy a központi csúcs címkéje legyen 0, míg az első körön az egyes számú címke szerepeljen, a második körön pedig a legnagyobb címke szerepeljen. Tehát ebben az esetben is létezik graceful címkézés. Ha a csúcsok számát növeljük, azt tapasztaljuk, hogy kis n -re létezik graceful címkézés különböző double-wheel gráfok esetén. Ugyanez mondható ki, triple-wheel gráfok esetén is.

4.2. Graceful caterpillar gráf konstrukciója

A második gráf osztályunk, azok a caterpillar gráfok lesznek. A program készítése során elsőként a caterpillar gráfnál használt központi utat konstruáltam meg, majd a program a felhasználó által beírt számok alapján, hozzákapcsol annyi levelet a csúcsokhoz, ahová és amennyit a felhasználó szeretne.

4.4. ábra. Caterpillar gráf adott levelekkel



```
def CaterpillarGraph(T):
    G = Graph()
    G.add_path(range(len(T))[0:len(T)])
    vc=len(G.get_vertices())
    for i in range(len(T)):
        for j in range(T[i]):
            G.add_edge(i,vc)
            vc=vc+1
    return G
```

A címkézés, az előző fejezetben található (Minden caterpillar gráf graceful) tétel bizonyítása szerint történik, ahol konstrukció alapján bizonyítottuk be állításunkat.

```
def CaterpillarLabels(T):
    under=-1
    over=-1
    leaves=[]
```

```

mainpath=[]
for i in range(len(T)):
    over=over+1+T[i]
join=over
for i in range(len(T)):
    mainpath.insert(len(mainpath),join)
    if i % 2 != 0:
        for j in range(T[i]):
            over=over-1
            leaves.insert(len(leaves),over)
        over=over-1
        join=over
    else:
        for j in range(T[i]):
            under=under+1
            leaves.insert(len(leaves),under)
        under=under+1
        join=under
return mainpath+leaves

```

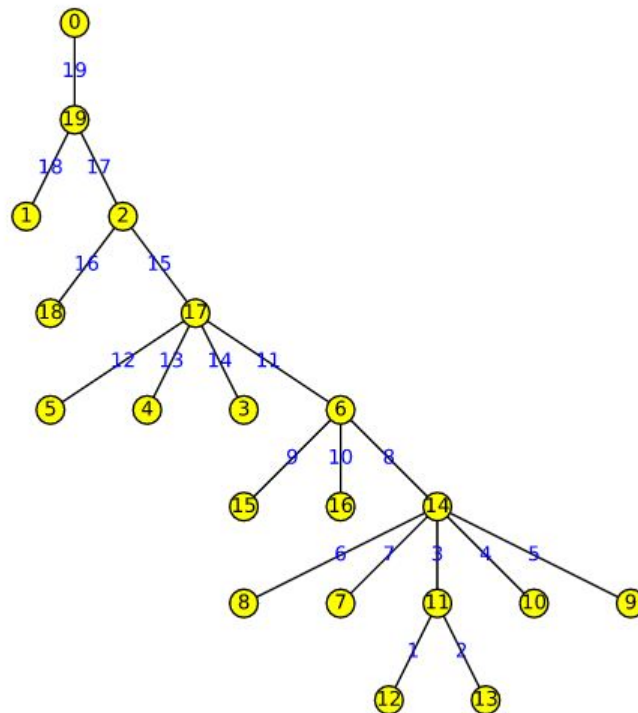
Szintén, mint a double-wheel gráfok során láthattuk, az alábbi programkód annyit csinál, hogy kirajzoltatja a gráfot, a tömbben szereplő címkékkel felcímkézi, és láthatóvá teszi nekünk a gráfot. A `D.show()` résznél rengeteg lehetőségünk van a gráfokat szerkeszteni, meg lehet adni a csúcsok és élek színét, vagy hogy hogyan helyezkedjen el a gráf.

```

def Caterpillar(N):
    D=CaterpillarGraph(N)
    T=CaterpillarLabels(N)
    D.relabel( [ T[i] for i in range(D.order()) ], inplace=True)
    for u,v,l in D.edges(): D.set_edge_label(u,v,abs(u-v))
    D.show(edge_labels=True,layout='tree',tree_orientation='down',
           tree_root=0,
           vertex_colors='yellow',figsize=7)
Caterpillar([2,7,2,2,2,3])

```

4.5. ábra. Caterpillar gráf címkézéssel



4.3. Graceful n-szeresen összetett teljes páros gráf konstrukciója

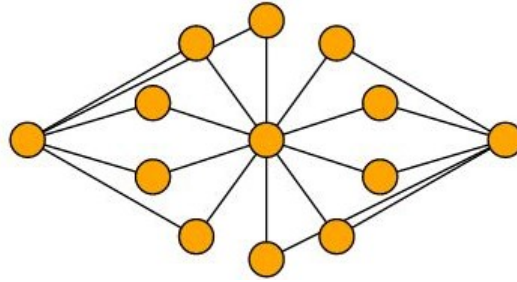
Most vizsgáljuk meg, hogy páros gráfok esetén, hogy alakul a címkézés. A konstrukció során, nem csak a páros gráfokat tudjuk leprogramozni, úgy hogy gracefully legyenek, hanem egy olyan gráfot is tudunk készíteni vele, aminek van egy központi csúcsa, és ez a csúcs az összes adott teljes páros gráfnak az egyik csúcsa. Ezt a gráf típust nevezzük n -szeren összetett teljes páros gráfnak.

```
def BigArrayOk(T):
    if len(T) == 0:
        return True
    if len(T) > 1:
        return False
    if T[0][1] > 2:
        return False
    return True

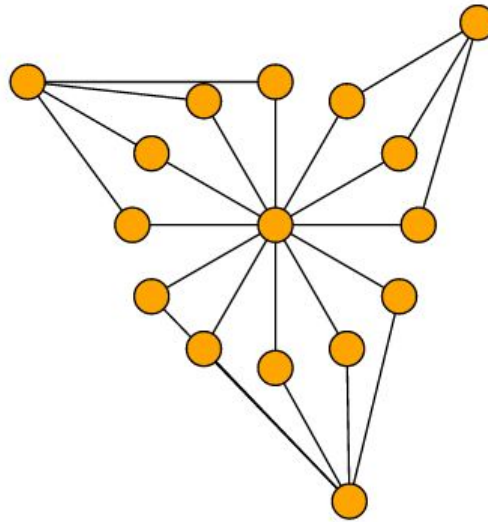
def ArraySort(T):
    for i in range(1, len(T)):
        val = T[i]
```

```
        j = i - 1
        while (j >= 0) and (T[j] < val):
            T[j+1] = T[j]
            j = j - 1
        T[j+1] = val
    return T
def ArrayDupes(T):
    count_dict = {}
    dupes=[]
    for i in range(len(T)):
        count_dict[T[i]] = 0
    for i in range(len(T)):
        count_dict[T[i]] = count_dict[T[i]] + 1
    for i in count_dict:
        if count_dict[i] > 1:
            dupes.insert(len(dupes), [i, count_dict[i]])
    return dupes
def ArraySum(T):
    a=0
    for i in range(len(T)):
        a=a+T[i]
    return a
```

4.6. ábra. 2-szeresen összetett teljes páros gráf



4.7. ábra. 3-szorosan összetett teljes páros gráf



A konstrukció során, minden ilyen jellegű gráfot meg lehet alkotni, viszont lesznek olyan gráfok, amik nem lesznek graceful tulajdonságúak. Ahhoz, hogy legyen graceful címkézése a gráfnak, ahhoz az kell, hogy a teljes páros gráfok közül maximum kettő legyen izomorf egymással. Ha több ilyen izomorf gráfot is összekapcsolunk, vagy 3-at, vagy 2-2-t, akkor a gráfot nem lehet graceful címkékkel ellátni.

```
def BigGraphLabels(T):
    L=[]
    dupes=ArrayDupes(T)
    if BigArrayOk(dupes):
        print("Van lehetőség a címkézésre!")
        if len(dupes) < 1:
            dupes.insert(len(dupes),[-1,-1])
        lastedge=0
```

```

        L.insert(len(L),0)
        i=0
        while i < len(T):
            L.insert(len(L),T[i])
            if T[i] == dupes[0][0]:
                L.insert(len(L),1)
                i = i + 1
            i = i + 1
        i=0
        while i < len(T):
            lastedge = lastedge+T[i]+1
            for j in range(T[i]):
                L.insert(len(L),lastedge+j)
            if T[i] == dupes[0][0]:
                lastedge = lastedge+T[i]+1
                for j in range(T[i]):
                    L.insert(len(L),lastedge+(2*j))
                i = i + 1
            i = i + 1
            lastedge=L[len(L)-1]
        else:
            print("Nincs lehetőség a címkézésre!")
            for i in range(ArraySum(T)+PartSum(T,0)):
                L.insert(len(L),i)
        return L
def PosDictCalc(T,L):
    R=20
    r=10
    n=ArraySum(T)
    pos_dict = {}
    pos_dict[L[0]]=[0,0]
    m=0.0
    for i in range(len(T)):
        m=m+T[i]/2
        x = R*float(cos(((2*pi)/n)*m))
        y = R*float(sin(((2*pi)/n)*m))
        pos_dict[L[i+1]] = [x,y]
        m=m+T[i]/2
    for i in range(n):
        x = r*float(cos(((2*pi)/n)*i))
        y = r*float(sin(((2*pi)/n)*i))

```

```

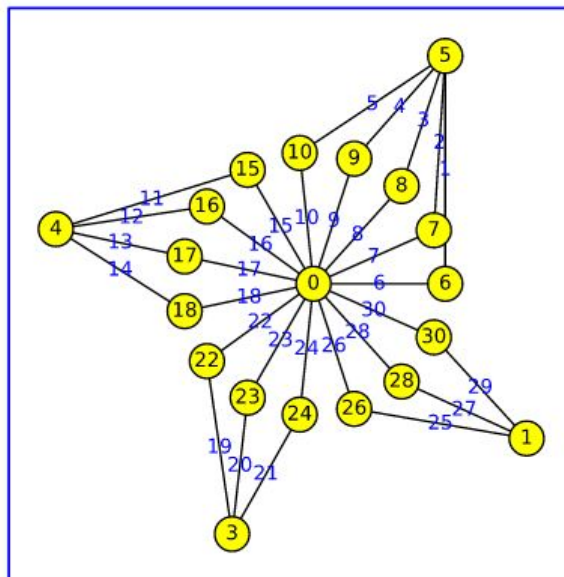
        pos_dict[L[i+PartSum(T,0)]] = [x,y]
    return pos_dict
def PartSum(T,a):
    n=len(T)+1
    for i in range(a):
        n=n+T[i]
    return n
def BipartGraf(L):
    G = Graph(len(L)+1)
    for i in range(len(L)):
        k=PartSum(L,i)
        for j in range(L[i]):
            G.add_edge(0,k+j)
            G.add_edge(i+1,k+j)
    return G
def Bipart(L):
    L=ArraySort(L)
    D = BipartGraf(L)
    Labels=BigGraphLabels(L)
    D.relabel( [ Labels[i] for i in range(D.order()) ], inplace=True)
    for u,v,l in D.edges(): D.set_edge_label(u,v,abs(u-v))
    pos_dict=PosDictCalc(L,Labels)
    pl = D.graphplot(pos=pos_dict,edge_labels=True,
        edge_color='black',vertex_colors='yellow',
        ,vertex_size=300,graph_border=True)
    pl.show(figsize=8)
Bipart([4,3,3,3])

```

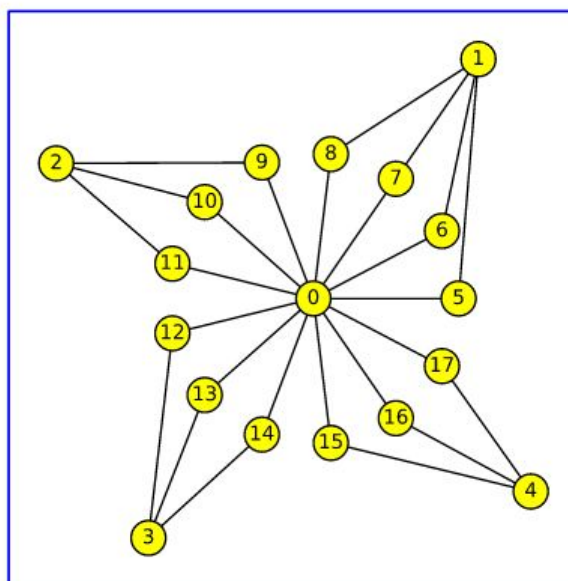
Ebben a programrészben, a címkézési eljárás látható, illetve az átláthatóság miatt a pozíció is meg lett adva. Ha olyan gráfot adunk meg, amiben több izomorf gráf is összekapcsolódik, azoknál az eseteknél is kirajzolja a gráfot, de kapunk egy üzenetet, hogy a gráfhoz nincs lehetőség graceful címkézésre.

4.8. ábra. Címkezés sikeres, illetve sikertelen esetben

Van lehetőség a címkezésre!



Nincs lehetőség a címkezésre!



4.4. Graceful pendant gráf konstrukciója

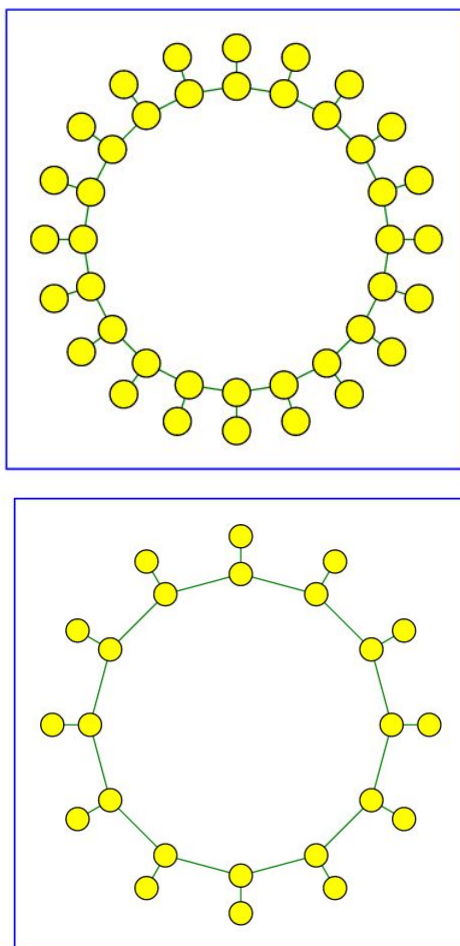
Az utolsó gráf, amit szemléltetni fogok, az a pendant gráf, ahol a pendant magyarul medált jelent. Nevét a gráf kinézetéről kapta, ami egy medálhoz hasonlít, de itt is, mint a többi esetben, nem medál gráfként fogom nevezni, hanem maradok a pendant gráfnál.

4.4.1. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = 2 \cdot n$. A G gráfot pendant gráfnak nevezzük, ha egy zárt út minden csúcsához, pontosan egy csúcs csatlakozik, amikből más él nem húzható.

4.4.2. Definíció. Legyen $G = (V, E, \varphi)$ véges gráf, $|V(G)| = 2 \cdot n$. A G pendant gráf gracefully címkézése:

1. elsőfajú, ha a 0 . és a $2 \cdot n$. címkék a zárt úton helyezkednek el
2. másodfajú, ha a $2 \cdot n$. címke szomszédos a 0 . címkével és a 0 . címke a zárt úton van, a $2 \cdot n$. címke pedig nincs az úton
3. harmadfajú, ha a $2 \cdot n$ címke a zárt úton van és szomszédja a 0 csúcs nincs a zárt úton.

4.9. ábra. 20 és 12 csúcsú pendant gráf



4.4.1. Tétel. Legyen $f : \mathbb{N} \rightarrow \mathbb{N} \cup 0$, és legyen $G = (V, E, \varphi)$ véges pendant gráf, ahol a zárt út csúcsai a $\{v_1, \dots, v_n\}$ halmazból kerülnek ki, míg a maradék csúcsok a $\{u_i, \dots, u_n\}$ halmazból kerülnek ki. Ha az u_i és v_j szomszédosak, akkor $i = j$ -vel.

Ha $n \equiv 4 \pmod{8}$ és $n > 12$, akkor a G gráfnak van graceful címkézése, ami elsőfajú és a következőképpen adható meg:

$$f(v_i) = \begin{cases} 0 & , \text{ ha } i = n \\ 2 \cdot n - i + 1 & , \text{ ha } i \text{ páratlan és } 1 \leq i \leq n - 1 \\ i + 2 & , \text{ ha } i = \frac{n}{2}, \frac{n}{2} + 2, \dots, n - 2 \\ i & , \text{ ha } i \text{ páros és } 2 \leq i \leq \frac{n}{2} - 2. \end{cases}$$

$$f(u_j) = \begin{cases} 2 \cdot n - 1 & , \text{ ha } j = n \\ 1 & , \text{ ha } j = n - 1 \\ \frac{n}{2} + 1 & , \text{ ha } j = n - 2 \\ 2 \cdot n - i + 1 & , \text{ ha } j = \frac{3 \cdot n}{4} + 1, \frac{3 \cdot n}{4} + 3, \dots, n - 4 \\ i + 6 & , \text{ ha } j = \frac{3 \cdot n}{4}, \frac{3 \cdot n}{4} + 2, \dots, n - 3 \\ i + 5 & , \text{ ha } j = \frac{3 \cdot n}{4} - 1 \\ i + 4 & , \text{ ha } j = \frac{n}{2} - 1, \frac{n}{2} + 1, \dots, \frac{3 \cdot n}{4} - 2 \\ 2 \cdot n - i - 1 & , \text{ ha } j \text{ páros és } 2 \leq j \leq \frac{3 \cdot n}{4} - 3 \\ i + 2 & , \text{ ha } j \text{ páratlan és } 1 \leq j \leq \frac{n}{2} - 3 \end{cases}$$

4.4.2. Tétel. Legyen $f : \mathbb{N} \rightarrow \mathbb{N} \cup 0$, és legyen $G = (V, E, \varphi)$ véges pendant gráf, ahol a zárt út csúcsai a $\{v_1, \dots, v_n\}$ halmazból kerülnek ki, míg a maradék csúcsok a $\{u_i, \dots, u_n\}$ halmazból kerülnek ki. Ha az u_i és v_j szomszédosak, akkor $i = j$ -vel.

Ha $n \equiv 3 \pmod{16}$ és $n > 19$, akkor a G gráfnak van graceful címkézése, ami elsőfajú és a következőképpen adható meg:

$$f(v_i) = \begin{cases} 0 & , \text{ ha } i = n \\ 2 \cdot n - i + 1 & , \text{ ha } i \text{ páratlan és } 1 \leq i \leq n - 2 \\ i + 2 & , \text{ ha } i = \frac{n+1}{2}, \frac{n+1}{2} + 2, \dots, n - 1 \\ i & , \text{ ha } i \text{ páros és } 2 \leq i \leq \frac{n+1}{2} - 2. \end{cases}$$

$$f(u_j) = \begin{cases} 2 \cdot n - 1 & , \text{ ha } j = n \\ 1 & , \text{ ha } j = n - 1 \\ \frac{n+1}{2} + 1 & , \text{ ha } j = n - 2 \\ n + 2 & , \text{ ha } j = n - 3 \\ 2 \cdot n - i - 1 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} \\ 2 \cdot n - i + 2 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 1 \\ 2 \cdot n - i - 7 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} - 2, \frac{3 \cdot n - 1}{4} + 2, \dots, n - 7 \\ 2 \cdot n - i + 1 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 4, \frac{3 \cdot n - 1}{4} + 8, \dots, n - 5 \\ i + 2 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 3, \frac{3 \cdot n - 1}{4} + 5, \dots, n - 4 \\ 2 \cdot n - i - 1 & , \text{ ha } j \text{ páros és } 2 \leq j \leq \frac{3 \cdot n - 1}{4} - 4 \\ i + 4 & , \text{ ha } j = \frac{n+1}{2} - 1, \frac{n+1}{2} + 1, \dots, \frac{3 \cdot n - 1}{4} - 1 \\ i + 2 & , \text{ ha } j \text{ páratlan és } 1 \leq j \leq \frac{n+1}{2} - 3 \end{cases}$$

4.4.3. Tétel. Legyen $f : \mathbb{N} \rightarrow \mathbb{N} \cup 0$, és legyen $G = (V, E, \varphi)$ véges pendant gráf, ahol a zárt út csúcsai a $\{v_1, \dots, v_n\}$ halmazból kerülnek ki, míg a maradék csúcsok a $\{u_i, \dots, u_n\}$ halmazból kerülnek ki. Ha az u_i és v_j szomszédosak, akkor $i = j$ -vel.

Ha $n \equiv 11 \pmod{16}$ és $n > 27$, akkor a G gráfnak van gracefully címkézése, ami elsőfajú és a következőképpen adható meg:

$$f(v_i) = \begin{cases} 0 & , \text{ ha } i = n \\ 2 \cdot n - i + 1 & , \text{ ha } i \text{ páratlan és } 1 \leq i \leq n - 2 \\ i + 2 & , \text{ ha } i = \frac{n+1}{2}, \frac{n+1}{2} + 2, \dots, n - 1 \\ i & , \text{ ha } i \text{ páros és } 2 \leq i \leq \frac{n+1}{2} - 2. \end{cases}$$

$$f(u_j) = \begin{cases} 2 \cdot n - 1 & , \text{ ha } j = n \\ 1 & , \text{ ha } j = n - 1 \\ \frac{n+1}{2} + 1 & , \text{ ha } j = n - 2 \\ n + 4 & , \text{ ha } j = n - 3 \\ n & , \text{ ha } j = n - 5 \\ 2 \cdot n - i - 1 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} \\ 2 \cdot n - i + 2 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 1 \\ 2 \cdot n - i - 7 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} - 2, \frac{3 \cdot n - 1}{4} + 2, \dots, n - 9 \\ 2 \cdot n - i + 1 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 4, \frac{3 \cdot n - 1}{4} + 8, \dots, n - 7 \\ i + 2 & , \text{ ha } j = \frac{3 \cdot n - 1}{4} + 3, \frac{3 \cdot n - 1}{4} + 5, \dots, n - 4 \\ 2 \cdot n - i - 1 & , \text{ ha } j \text{ páros és } 2 \leq j \leq \frac{3 \cdot n - 1}{4} - 4 \\ i + 4 & , \text{ ha } j = \frac{n+1}{2} - 1, \frac{n+1}{2} + 1, \dots, \frac{3 \cdot n - 1}{4} - 1 \\ i + 2 & , \text{ ha } j \text{ páratlan és } 1 \leq j \leq \frac{n+1}{2} - 3 \end{cases}$$

Ezen tételek bizonyítását a [2] folyóiratban olvashatjuk, de a tételeket felhasználva a Sage programcsomag segítségével el lehet készíteni egy olyan programot, mely mind a három tétel feltételeit tartalmazza és elkészíti a címkét,

majd kirajzoltatja a pendant gráfot. Ha olyan esetet adunk meg, mely egyik tételt se tudja alkalmazni, úgy egy hibaüzenetet kapunk.

```
def Labels(n):
    L=[]
    if n % 8 == 4 and n>=12:
        for i in range(1,n+1):
            if i == n:
                L.insert(len(L),0)
            elif i % 2 == 0 and i<(n+1)/2-2:
                L.insert(len(L),i)
            elif i % 2 <> 0 and i<n:
                L.insert(len(L),2*n-i+1)
            elif n/2<=i<=n-2 and i % 2 == 0:
                L.insert(len(L),i+2)
        for i in range(1,n+1):
            if i == n:
                L.insert(len(L),2*n-1)
            elif i == n-1:
                L.insert(len(L),1)
            elif i == n-2:
                L.insert(len(L),n/2+1)
            elif i == 3*n/4-1:
                L.insert(len(L),i+5)
            elif 3*n/4+1<=i<=n-3 and i % 2 == 0:
                L.insert(len(L),2*n-i+1)
            elif 3*n/4<=i<=n-2 and i % 2 <> 0:
                L.insert(len(L),i+6)
            elif n/2-1<=i<=3*n/4-1 and i % 2 <> 0:
                L.insert(len(L),i+4)
            elif 2<=i<=3*n/4-2 and i % 2 == 0:
                L.insert(len(L),2*n-i-1)
            elif 1<=i<=n/2-2 and i % 2 <> 0:
                L.insert(len(L),i+2)
    elif n % 16 == 3 and n>=19:
        for i in range(1,n+1):
            if i == n:
                L.insert(len(L),0)
            elif i % 2 == 0 and i<=(n+1)/2-2:
                L.insert(len(L),i)
            elif i % 2 <> 0 and i<=n-2:
                L.insert(len(L),2*n-i+1)
```

```

        elif (n+1)/2<=i<=n-1 and i % 2 == 0:
            L.insert(len(L),i+2)
    for i in range(1,n+1):
        if i == n:
            L.insert(len(L),2*n-1)
        elif i == n-1:
            L.insert(len(L),1)
        elif i == n-2:
            L.insert(len(L),(n+1)/2+1)
        elif i == n-3:
            L.insert(len(L),n+2)
        elif i == (3*n-1)/4:
            L.insert(len(L),2*n-i-1)
        elif i == (3*n-1)/4+1:
            L.insert(len(L),2*n-i+2)
        elif (3*n-1)/4-2<=i<=n-7 and i % 2 == 0:
            L.insert(len(L),2*n-i-7)
        elif (3*n-1)/4+4<=i<=n-5 and i % 2 == 0:
            L.insert(len(L),2*n-i+1)
        elif (3*n-1)/4+3<=i<=n-4 and i % 2 <> 0:
            L.insert(len(L),i+2)
        elif 2<=i<=(3*n-1)/4-4 and i % 2 == 0:
            L.insert(len(L),2*n-i-1)
        elif (n+1)/2-1<=i<=(3*n-1)/4-1 and i % 2 <> 0:
            L.insert(len(L),i+4)
        elif 1<=i<=(n+1)/2-3 and i % 2 <> 0:
            L.insert(len(L),i+2)
    elif n % 16 == 11 and n>=27:
        for i in range(1,n+1):
            if i == n:
                L.insert(len(L),0)
            elif i % 2 == 0 and i<=(n+1)/2-2:
                L.insert(len(L),i)
            elif i % 2 <> 0 and i<=n-2:
                L.insert(len(L),2*n-i+1)
            elif (n+1)/2<=i<=n-1 and i % 2 == 0:
                L.insert(len(L),i+2)
        for i in range(1,n+1):
            if i == n:
                L.insert(len(L),2*n-1)
            elif i == n-1:

```

```

        L.insert(len(L),1)
    elif i == n-2:
        L.insert(len(L),(n+1)/2+1)
    elif i == n-3:
        L.insert(len(L),n+4)
    elif i == n-5:
        L.insert(len(L),n)
    elif i == (3*n-1)/4:
        L.insert(len(L),2*n-i-1)
    elif i == (3*n-1)/4+1:
        L.insert(len(L),2*n-i+2)
    elif (3*n-1)/4-2<=i<=n-9 and i % 2 == 0:
        L.insert(len(L),2*n-i-7)
    elif (3*n-1)/4+4<=i<=n-7 and i % 2 == 0:
        L.insert(len(L),2*n-i+1)
    elif (3*n-1)/4+3<=i<=n-4 and i % 4 <> 0 and i % 2 <> 0:
        L.insert(len(L),i+2)
    elif 2<=i<=(3*n-1)/4-4 and i % 2 == 0:
        L.insert(len(L),2*n-i-1)
    elif (n+1)/2-1<=i<=(3*n-1)/4-1 and i % 2 <>0:
        L.insert(len(L),i+4)
    elif 1<=i<=(n+1)/2-3 and i % 2 <> 0:
        L.insert(len(L),i+2)

    return L

def PendantGraph(n):
    G=Graph()
    G.add_vertices(range(2*n))
    for i in range(n):
        G.add_edge(i,n+i)
        if i != 0:
            G.add_edge(i-1,i)
        else:
            G.add_edge(n-1,i)
    return G

```

A címkézés után, a következő programkód a pendant gráfot alkotja meg. Lényege, hogy tetszőlegesen vesz fel $2 \cdot n$ darab csúcsot, majd éllel köti össze őket, hogy egy zárt utat kapjunk, és a maradék csúcsokat pedig különböző úton található csúcsokhoz köti éllel.

```

def PendantGraph(n):
    G=Graph()

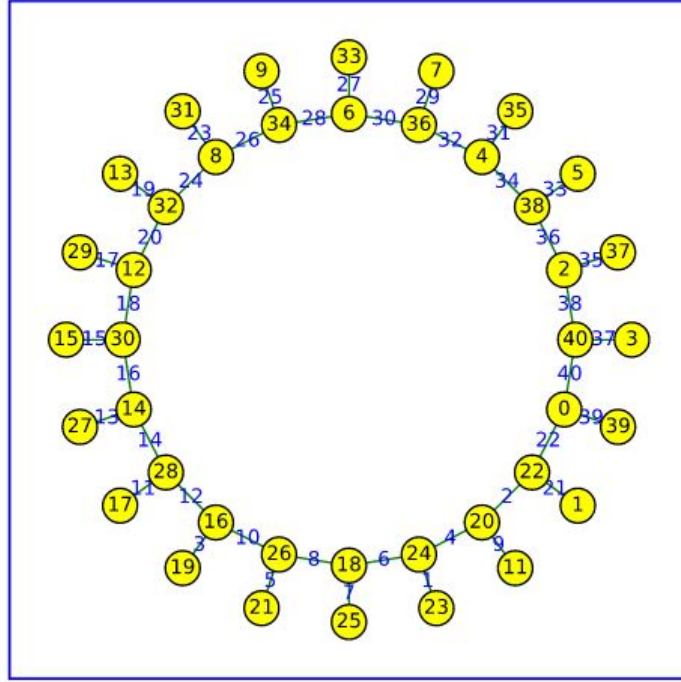
```

```
G.add_vertices(range(2*n))
for i in range(n):
    G.add_edge(i,n+i)
    if i != 0:
        G.add_edge(i-1,i)
    else:
        G.add_edge(n-1,i)
return G
def Pendant(n):
    F=PendantGraph(n)
    L=Labels(n)
    draw(F, L)
```

Végül R1 és R2 segítségével, megadjuk az elhelyezkedését a csúcsoknak, úgy hogy R1 és R2 jelenti a kör sugarát.

```
def draw(G, L):
    R1=20
    R2=25
    pos_dict = {}
    for i in range(len(L)/2):
        x=R1*float(cos(((2*pi)/(len(L)/2))*i))
        y=R1*float(sin(((2*pi)/(len(L)/2))*i))
        pos_dict[L[i]] = [x,y]
    for i in range(len(L)/2,len(L)):
        x=R2*float(cos(((2*pi)/(len(L)/2))*i))
        y=R2*float(sin(((2*pi)/(len(L)/2))*i))
        pos_dict[L[i]] = [x,y]
    G.relabel( [ L[i] for i in range(G.order()) ], inplace=True)
    for u,v,l in G.edges(): G.set_edge_label(u,v,abs(u-v))
    pl = G.graphplot(pos=pos_dict,edge_labels=True,
        edge_color='green',vertex_colors='yellow',
        vertex_size=300,graph_border=True)
    pl.show(figsize=7)
Pendant(20)
```

4.10. ábra. 20 csúcsú pendant gráf graceful címkézése



A tételek érdekességei az, ha az ember megfigyeli, hogy mi történik az $n = 12$, $n = 19$ és $n = 27$ esetekben, akkor ha leprogramozza szépen a feltételeket, akkor észreveheti, hogy a tétel azokban az esetben is helytálló. Azonban, ez nem lehetséges, mert a számítógép nem veszi észre, hogy a feltételek ütköznek, így lesz olyan i érték, amire két érték is helyes. Ezt azonban a számítógép nem írja felül, és csak az első értéket adja vissza. Például az első tétel esetében, az $n = 12$ esetén, $i = n - 2$ és $i = \frac{3 \cdot n}{4} + 1$ is lehetséges. Mivel a számítógép ugye az első feltételt nézi meg, az $n = 12$ esetén $\frac{12}{2} + 1 = 7$ címkét kapjuk meg.

IRODALOMJEGYZÉK

- [1] Ronan Le Bras, Carla P. Gomes, and Bart Selman: *Double-Wheel Graphs Are Graceful*, Computer Science Department, Cornell University, United States of America 2013.
<http://www.cs.cornell.edu/~lebras/publications/LeBras2013Double.pdf>.
- [2] Alessandra Graf: *A new graceful labeling for pendant graphs*, Aequat.math, vol. 87, pp. 135-145, 2014.
<http://link.springer.com/article/10.1007/s00010-012-0184-4>.
- [3] Aldred, R.E.L.; McKay, Brendan D. : *Graceful and harmonious labellings of trees.*, Bulletin of the Institute of Combinatorics and its Applications 23: 69-72, 1998.
<http://users.cecs.anu.edu.au/~bdm/papers/graceful.pdf>
- [4] Gallian, Joseph A.: *A dynamic survey of graph labeling*, Electronic Journal of Combinatorics 5: Dynamic Survey 6, 43 pp. , 1998.
<http://www.combinatorics.org/ojs/index.php/eljc/article/viewFile/DS6/pdf>
- [5] Charles Michael Cavalier: *Graceful labelings*, Louisiana State University, United States of America, 2006.
<http://people.math.sc.edu/czabarka/Theses/Graceful.pdf>
- [6] G. Sethuraman: *On graceful graphs: One vertex unions of non-isomorphic complete bipartite graphs*, Indian J. Pure Appl. Math. 32, No.7, 975-980 , 2001.
[http://www.dli.gov.in/data/_\\$copy/upload/INSA/INSA\\$_\\$1/20005b06\\$_\\$975.pdf](http://www.dli.gov.in/data/_$copy/upload/INSA/INSA$_$1/20005b06$_$975.pdf)
- [7] Elina Robeva: *An extensive survey of graceful trees*, [Undergraduate Honours thesis], Stanford University, United States of America 2011.
<http://mathematics.stanford.edu/wp-content/uploads/2013/08/Robeva-Honors-Thesis-2011.pdf>
- [8] W.A. Stein et al.: *Sage Mathematics Software (Version 7.1)*, The Sage Development Team, 2016, <http://www.sagemath.org>.

- [9] Turjányi Sándor: *Bevezetés a kombinatorikába és a gráfelméletbe*, mobiDIÁK könyvtár, Debreceni Egyetem, Informatikai Kar, 2005.
http://www.inf.unideb.hu/oktatas/mobidiak/Turjanyi_Sandor/Bev._a_komb._es_a_grafelméletbe/1.pdf
- [10] Hajnal Péter: *Gráfelmélet*, Polygon, Szeged, 1997.
- [11] Rosa, A.: *On certain valuations of the vertices of a graph*. Theory of Graphs (Internat. Symposium, Rome, July 1966), pp. 349-355. Gordon and Breach, New York; Dunod, Paris, 1967.
https://www.researchgate.net/publication/244474213_On_certain_valuations_of_the_vertices_of_a_graph_Theory_of_Graphs
- [12] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, G. Tack.: *Towards a Standard CP Modelling Language*. In Bessière (ed.), *Thirteenth International Conference on Principles and Practice of Constraint Programming*. Lecture Notes in Computer Science 4741, pp. 529–543. Springer-Verlag, Providence, RI, USA, Sept. 2007.
<http://www.gecode.org/paper.html?id=NethercoteStuckeyEa:CP:2007>