1. Grafteori

Betydelsen av bilder som stöd och inspiration för matematiska resonemang kan knappast överskattas. Studierna av enkla bilder har gett oss grafteorin. Tyvärr, eller lyckligtvis, visar det sig snabbt att enkla och naturliga frågeställningar om enkla bilder leder till synnerligen komplicerade kombinatoriska resonemang.

I detta inledande kapitel om grafteori kommer vi att nöja oss med grundläggande terminologi och några resultat som är enkla att bevisa.

Grafer

Definition. *En Graf (V, E) består av två mängder V och E av noder (ibland "hörn", vertex på engelska) respektive bågar (ibland "kanter", edge på engelska), där varje båge $e \in E$ går mellan två olika noder i V, bågens ändpunkter, och där det inte finns flera bågar med samma ändpunkter.*

Noder ritas som feta punkter (ibland namngivna). Om (V, E) är en graf och $e \in E$ är en båge med ändpunkterna $a, b \in V$, så identifieras bågen med mängden $\{a, b\} = \{b, a\} = e$.

Exempel.

Figur 1. Två grafer. Till vänster en graf med mängden av noder $V = \{a, b, c, d\}$ och mängden av bågar $E = \{\{a, b\}, \{a, d\}, \{c, d\}\}$.

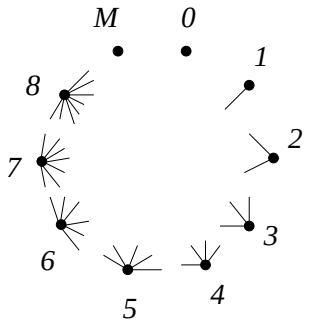
Figur 2. Exempel på multigrafer som inte är grafer.

Definition. *I en multigraf tillåter vi mer än en båge mellan ett par av noder, samt bågar från en nod till samma nod.*

Speciellt noterar vi att varje graf är en multigraf.

Exempel. Herr McBrain och hans fru April ordnar en fest och bjuder fyra andra gifta par. En del av människorna hälsar på varandra då de träffas, men naturligtvis hälsar inget par på varandra. Då festen slutar frågar Herr McBrain alla andra, på hur många människor de har hälsat och han erhåller 9 olika svar. Hur många människor hälsade på April?

Lösning: Vi konstruerar en graf vars noder är människor på festen och det finns en båge $\{a, b\}$ då a och b har hälsat på varandra. Emedan det finns 9 människor utöver Herr McBrain och maximala antalet hälsningar i vilket en person kan vara involverad i är 8, så följer att de olika svaren som Herr McBrain fick måste vara $0, 1, 2, 3, 4, 5, 6, 7, 8$. Vi betecknar noderna med dessa siffror och använder M för att beteckna Herr McBrain. Vi får följande bildrepresentation av grafen:



Figur 3.

Noden 8 sammanbinds med alla andra noder utom en, som måste vara äkta hälft till 8. Denna nod måste vara 0, varför 8 och 0 är gifta och 8 sammanbinds med $1, 2, ..., 7, M$. Speciellt gäller att 1 och 8 sammanbinds och detta är enda bågen utgående från 1. Således är 7 inte sammankopplad med 0 och 1, och därmed är 7 och 1 gifta, (emedan 0 och 8 är gifta). Fortsätter vi på detta sätt, ser vi att 6 och 2, samt 5 och 3 är gifta par. Det följer att M och 4 är gifta, värför noden 4 är April som hälsar på 4 människor.

Fastän bildrepresentationer av grafer förstås av människor, så är de värdelösa när vi önskar kommunicera med en dator. För detta ändamål måste vi representera en graf med någon typ av tabell, exempelvis en matrisrepresentation.

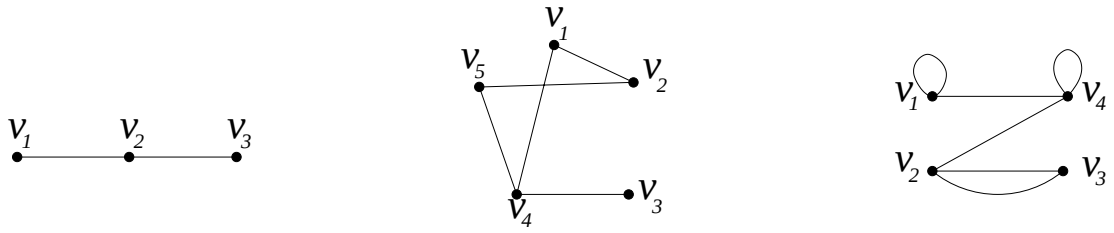
Definition. *Vi säger att två noder a och b i en graf är grannar om $\{a, b\}$ är en båge, dvs. om a och b förenas av en båge. Låt (V, E) vara en graf med $|V| = n$, (här och i fortsättningen betecknar $|B|$ antalet element i en mängd B). Låt $V = \{v_1, ..., v_n\}$ vara noderna. Grannmatrisen A för (V, E) är en $n \times n$ matris vars element*

$$A_{i,j} = \begin{cases} 1, & \text{om } v_i \text{ och } v_j \text{ är grannar;} \\ 0, & \text{annars.} \end{cases}$$

För alla i, j gäller det att $A_{i,j} = A_{j,i}$, varför A är symmetrisk med nollor på diagonalen.

Det finns variationer av ovanstående representation. Om (V, E) är en multigraf, så betecknar $A_{i,j}$ antalet bågar $\{v_i, v_j\}$. Även i detta fall är A symmetrisk men behöver ingalunda ha endast nollor på diagonalen, ty det kan finnas bågar som utgår från och slutar i samma nod.

Exempel.



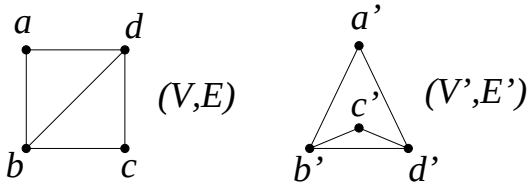
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Figur 4. Tre grafer samt deras grannmatriser.

Vi intresserar oss nu för frågeställningen: När är två grafer väsentligen lika? Svaret på denna fråga ges med hjälp av begreppet isomorfi (isos = samma, morph = form).

Definition. *Två grafer (V, E) och (V', E') är isomorfa om det existerar en bijektion $\varphi : V \to V'$ som avbildar bågar i E på bågar i E' och omvänt, dvs. $\{a, b\} \in E \Leftrightarrow \{\varphi(a), \varphi(b)\} \in E'$ för alla $a, b \in V$. Härvid kallas φ en isomorfism. Vi använder beteckningen $(V, E) \cong (V', E')$ för isomorfa grafer. (En bijektion är en avbildning som är surjektiv och injektiv).*

Exempel.



Figur 5.

$$\varphi : V \leftrightarrow\!\!\!\!\rightarrow V', \quad \varphi(a) = a', \quad \varphi(b) = b', \quad \varphi(c) = c', \quad \varphi(d) = d.'$$

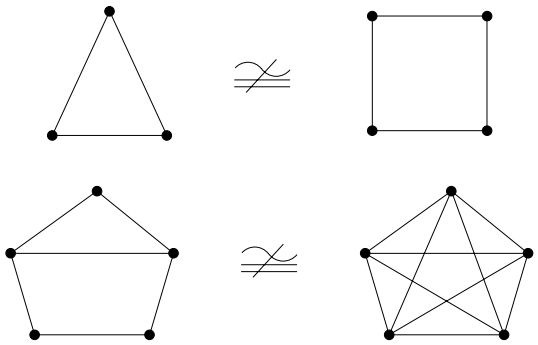
Alltså φ är en bijektion.

$$\{a, b\} \in E \Longleftrightarrow \{\varphi(a), \varphi(b)\} \in E'$$
$$\{a, d\} \in E \Longleftrightarrow \{\varphi(a), \varphi(d)\} \in E'$$
$$\{b, c\} \in E \Longleftrightarrow \{\varphi(b), \varphi(c)\} \in E'$$
$$\{b, d\} \in E \Longleftrightarrow \{\varphi(b), \varphi(d)\} \in E'$$
$$\{c, d\} \in E \Longleftrightarrow \{\varphi(c), \varphi(d)\} \in E'$$

Alltså φ är en isomorfism och $(V, E) \cong (V', E')$

För att visa att två grafer inte är isomorfa, måste vi visa att det inte finns någon bijektion från mängden av noder i den ena grafen till mängden av noder i den andra grafen som avbildar bågar på bågar. Om två grafer har olika antal noder, så finns det ingen bijektion och graferna kan inte vara isomorfa. Om graferna har samma antal noder men antalet bågar är olika, så finns det bijektioner mellan mängderna av noder men ingen av dessa bijektioner är en isomorfism.

Exempel.



Figur 6.

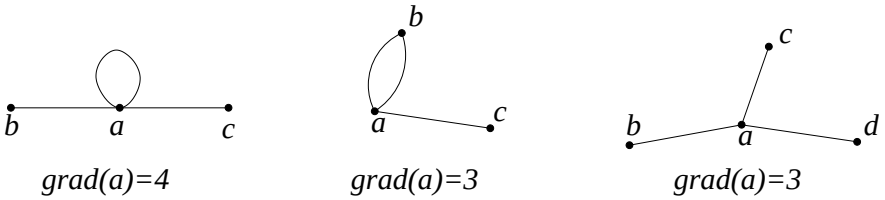
Det är i allmänhet mycket svårt att bedöma om två grafer är isomorfa, men vissa naturliga villkor kan vi ge som är nödvändiga för isomorfi. Vi återkommer till dessa villkor efter det att vi infört ett antal nya begrepp för grafer.

Definition. *Låt (V, E) vara en graf och antag att $a \in V$. Då är*

$$\text{grad}(a) = |\{b \in V : \{a, b\} \in E\}|$$

graden av a, dvs. graden av a är antalet bågar med a som ändpunkt. Analog definition för multigrafer, men en båge från en nod a till sig själv bidrar med två enheter då graden för a beräknas.

Exempel.



Figur 7.

Det är ofta av intresse att veta antalet bågar i en graf. De kan förstås räknas direkt men det är vanligtvis lättare att räkna antalet bågar vid varje nod och addera dem. Då har varje båge blivit medräknad två gånger, en gång vid båda ändpunkterna, så antalet bågar i en graf är halva denna summa.

Sats 1. *I en graf finns det ett jämnt antal noder av udda grad.*

Bevis: Låt (V, E) vara en graf med $V = \{a_1, ..., a_n\}$. Då gäller det att

$$|E| = \frac{1}{2} \sum_{i=1}^{n} \text{grad}(a_i)\,. \qquad (*)$$

Beteckna:

$$V_o = \{a_i \in V : \text{grad}(a_i) \text{ är udda}\},$$
$$V_e = \{a_i \in V : \text{grad}(a_i) \text{ är jämn}\}.$$

Då är $V = V_o \cup V_e$ och $V_o \cap V_e = \emptyset$, (en partition av V). Med stöd av $(*)$ erhålls:

$$2\,|E| = \sum_{a \in V_o} \text{grad}(a) + \sum_{a \in V_e} \text{grad}(a) \qquad (**)$$

Den andra summan i $(**)$ är jämn, ty varje term är jämn. Då $2\,|E|$ är ett jämnt tal, så måste även den första summan i $(**)$ vara jämn. Då den första summan är en summa av udda tal måste det då finnas ett jämnt antal termer i summan och därmed är satsen bevisad. □

Ett nödvändigt villkor för isomorfi är att

$$\text{grad}(a) = \text{grad}(\varphi(a)) \text{ för varje nod } a \in V.$$

Men detta villkor är inte tillräckligt, vilket framgår av följande exempel.

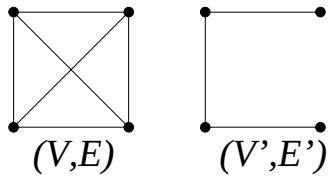
Exempel.

Figur 8.

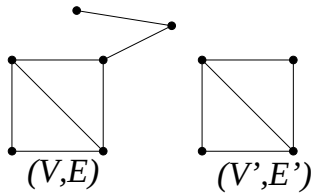
Antag att $\varphi : V \to V'$ är en isomorfism. Då 4 och 5 är de enda elementen i V och V' med grad(4) = grad(5) = 2, så gäller antingen ($\varphi(4) = 4,\ \varphi(5) = 5$) eller ($\varphi(4) = 5,\ \varphi(5) = 4$). Men eftersom $\{4,5\} \in E$ och $\{\varphi(4), \varphi(5)\} = \{4,5\} \notin E'$, så erhålls en motsägelse. Därmed finns det ingen isomorfism och $(V,E) \ncong (V',E')$.

Definition. *Grafen (V',E') är en delgraf av (V,E) om $V' \subseteq V$ och $E' \subseteq E$. Om $V' = V$, så är (V',E') en uppspännande delgraf av (V,E). Om $a,b \in V'$ och $\{a,b\} \in E \Rightarrow \{a,b\} \in E'$, så är delgrafen (V',E') en full delgraf av (V,E).*

Exempel.



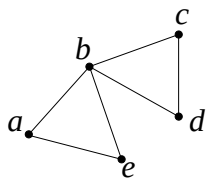
Figur 9. (V',E') är en uppspännande delgraf av (V,E).



Figur 10. (V',E') är en full delgraf av (V,E).

Definition. *En följd $a_0, a_1, ..., a_n$ av noder i (V,E) är en väg om $\{a_i, a_{i+1}\} \in E$ för $i = 0,1,\ldots,n \Leftrightarrow 1$, och $\{a_{i-1}, a_i\} \neq \{a_i, a_{i+1}\}$ för $i = 1, ..., n$-1 . Vägen sägs ha längden n. Om $a_0 = a_n$, så är vägen en loop. Om i loopen samtliga a_i, $i = 0, ..., n$-1, är olika, så är loopen en cykel. Om $a_0 \neq a_n$ kallas vägen öppen.*

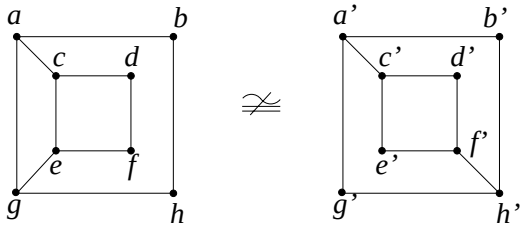
Exempel.



Figur 11. a, e, b, d, c, b är en (öppen) *väg* med längden 5. a, e, b, c, d, b, a är en *loop* (ej cykel). a, e, b, a är en *cykel*.

Om (V, E) och (V', E') är isomorfa grafer motsvaras vägarna i den ena grafen av vägar i den andra. Ett nödvändigt villkor för isomorfi är därför att antalet vägar av fix längd är lika i de båda graferna. Vidare har isomorfa grafer lika många cykler av fix längd.

Exempel.

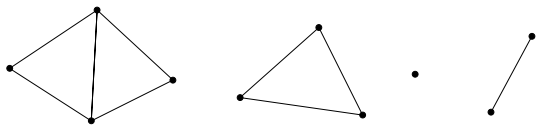


Figur 12. I den första grafen är $b, h, g, e, f, d, c, a, b$ en cykel av längden 8. Medan det i den andra grafen inte existerar någon cykel av längden 8. Detta innebär att graferna inte är isomorfa

Definition. *Låt (V, E) vara en graf. Två noder $a, b \in V$ är förenade om det finns en väg $a = a_0, a_1, ..., a_n = b$ i V. Om varje par av noder i grafen är förenade, så är grafen sammanhängande.*

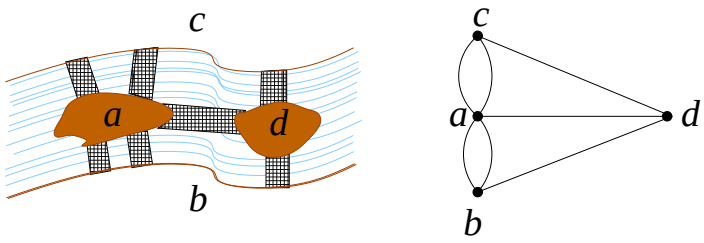
Om en graf G inte är sammanhängande kan vi inte nå alla noder med en väg från en given nod a. De noder som kan nås med en väg från en nod a och motsvarande bågar kallas en sammanhängande komponent av G. På detta sätt bildar de sammanhängande komponenterna en sönderdelning av grafen G. Två isomorfa grafer måste ha lika många sammanhängande komponenter.

Exempel.



Figur 13. En osammanhängande graf bestående av 4 sammanhängande komponenter.

Vi skall nu studera multigrafer. Som bekant, så tillåter vi för multigrafer mer än en båge mellan ett par av noder. Vi antar i fortsättningen att alla grafer och multigrafer är ändliga. Leonhard Euler (1707–1783, schweizare) kan kallas den första grafteoretikern. Han använde sig av begreppet multigraf för att lösa ett problem som invånarna i Königsberg (Kaliningrad) hade när de planerade sina söndagspromenader. Problemet var att finna en promenadväg som använde alla broar exakt en gång:

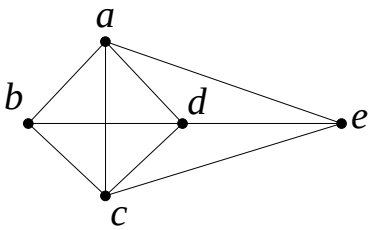


Figur 14. Sju broar, Euler ritade en multigraf.

Euler konstaterade att den sökta promenadvägen svarar mot en loop som använder sig av samtliga bågar i multigrafen. Begreppet väg för en multigraf definieras analogt med begreppet väg för en graf.

Definition. *En väg i en multigraf (V, E) är en Euler-väg om alla bågar i E används exakt en gång. Om $a_0 = a_n$ i en Euler-väg så har vi en Euler-loop.*

Exempel.



Figur 15. En *Euler-väg* ges av $b, a, c, b, d, a, e, c, d, e$ (2 noder med udda grad).

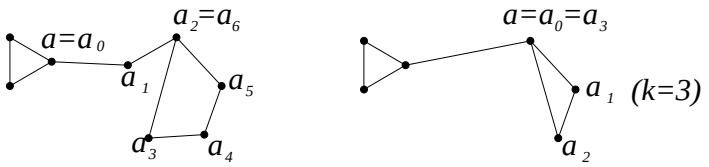
Anledningen till att ingen lyckats med att göra promenaden över broarna i Königsberg är:

Sats 2. *En sammanhängande ändlig multigraf (V, E) har en Euler-loop om och endast om varje nod har jämn grad.*

Vi noterar att i problemet gällande broarna i Königsberg så har varje nod udda grad. För beviset av Sats 2 behöver vi följande hjälpresultat.

Lemma 3. *Om (V, E) är en ändlig multigraf sådan att varje nod har grad ≥ 2 så finns det en cykel i grafen.*

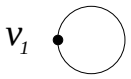
Bevis: (Av Lemma 3). Antag att (V, E) är en graf. (Om (V, E) är en multigraf som inte är en graf så är saken klar). Tag $a \in V$. Konstruera en väg $a = a_0, a_1, a_2, ...$ i grafen sådan att $\{a_0, a_1\} \in E$ och så att för $i \geq 1$ gäller att $\{a_i, a_{i+1}\} \in E$ och $a_{i+1} \neq a_{i-1}$. Detta lyckas eftersom $\text{grad}(a_i) \geq 2$ för alla i. Då nu V är en ändlig mängd finns det ett minsta k, $(k \geq 3)$, sådant att $a_k \in \{a_0, ..., a_{k-1}\}$. Antag att $a_k = a_j$, $0 \leq j \leq k$-3. Då är $a_j, a_{j+1}, ..., a_k$ en cykel. □



Figur 16.

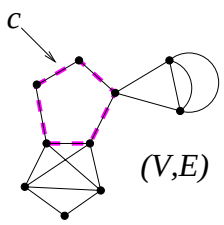
Bevis: (Av Sats 2). (1) Antag att (V, E) har en Euler-loop. Låt a vara en nod i loopen. Varje gång loopen passerar a förbrukas två olika bågar. Därmed är $\text{grad}(a)$ ett jämnt tal.

(2) Antag att varje nod har jämnt gradtal. (Induktion). Om $|E| = 1$ har vi fallet



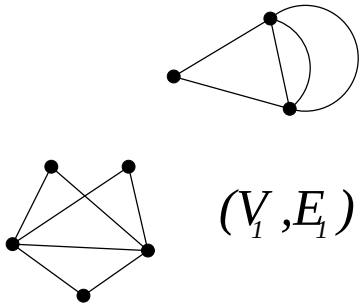
Figur 17. $grad(v_1) = 2$.

dvs. en Euler-loop. Antag att påståendet gäller för alla multigrafer (V', E') med $|E'| < |E|$ där $|E| \geq 2$. Då (V, E) är sammanhängande måste för varje $a \in V$ gälla att $\text{grad}(a) \geq 1$. Vidare antogs att varje nod har jämn grad så $\text{grad}(a) \geq 2$ för alla $a \in V$. Då ger Lemma 3 att det finns en cykel betecknad c i (V, E). Om c är en Euler-loop är vi klara. Annars gäller:



Figur 18.

Konstruera en delgraf (V_1, E_1) till (V, E) genom att först stryka bågarna i c och sedan noderna vars gradtal sjunkit till noll.



Figur 19.

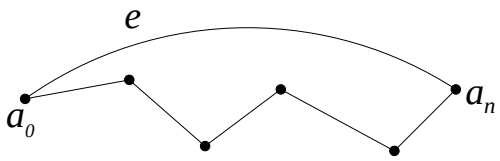
Samtliga noder i (V_1, E_1) har nu jämn grad. De sammanhängande komponenterna för (V_1, E_1) består av noder med jämnt gradtal och antalet bågar i varje sammanhängande komponent är mindre än $|E|$. Induktionsantagandet ger att varje komponent har en Euler-loop. Vi "skarvar ihop" c med Euler-looparna för alla komponenter och erhåller en Euler-loop för (V, E). □

Följande sats ger vid handen att Königsbergsborna inte ens har kunnat hitta en öppen Euler-väg.

Sats 4. *Låt (V, E) vara en sammanhängande ändlig multigraf. Då finns det en öppen Euler-väg om och endast om det finns exakt två noder med udda grad.*

Bevis: (1) Antag att a och b är de enda noderna av udda grad i (V, E). Bilda en ny båge e mellan a och b. Då ger Sats 2 att det finns en Euler-loop i grafen $(V, E \cup \{e\})$. Om vi avlägsnar bågen e får vi en öppen Euler-väg i (V, E).

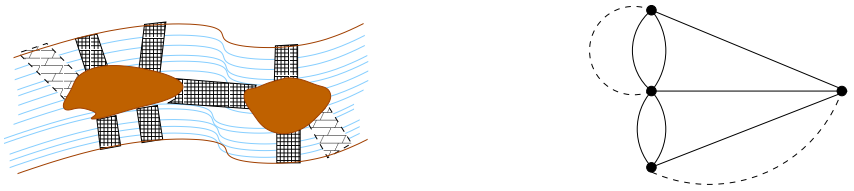
(2) Antag att det finns en öppen Euler-väg $a_0, ..., a_n$ i (V, E). Sätt $e = \{a_0, a_n\}$ och $E' = E \cup \{e\}$.



Figur 20.

Då finns det en Euler-loop i (V, E'). Sats 2 ger att alla noder i (V, E') är av jämn grad. Då har alla noder i (V, E), utom a_0 och a_n, jämn grad. □

Bygger vi en bro till i Kaliningrad kan vi hitta en Euler-väg och bygger vi två på lämpligt sätt kan vi erhålla en Euler-loop.



Figur 21.

Ett närliggande problem är följande: Givet en graf (V, E), avgör om det finns en loop i (V, E) som passerar varje nod exakt en gång.

Vi ser direkt att en sådan loop måste vara en cykel och om den existerar måste (V, E) vara sammanhängande.

Exempel.

Figur 22.

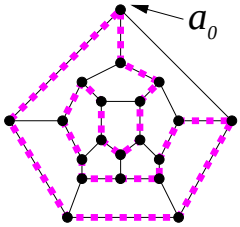
Figur 22 ger två exempel på grafer där det inte är möjligt att hitta en sådan cykel. Den som först intresserade sig för detta problem var irländaren W. Hamilton 1805–1865.

Definition. *En graf (V, E) kallas en Hamilton-graf om det existerar en cykel $a_0, ..., a_n, a_0$ sådan att $V = \{a_0, ..., a_n\}$.*

Det är i allmänhet ett mycket komplicerat problem att avgöra om en sammanhängande graf är en Hamilton-graf och det finns ännu ingen karakterisering av Hamilton-grafer.

Vanligen täcker inte en Hamilton-cykel alla bågar. Den täcker endast två bågar vid varje nod.

Exempel.

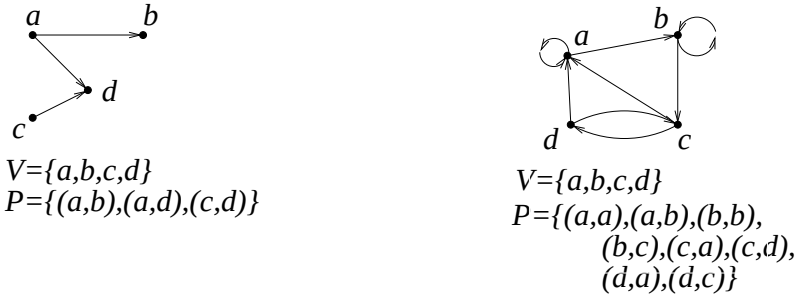


Figur 23.

En Euler-loop använder varje båge exakt en gång, medan en Hamilton-cykel besöker varje nod exakt en gång.

Definition. *En riktad graf (V, P) består av en mängd V av noder och en delmängd $P \subseteq V \times V$ vars element kallas pilar. Varje pil är en riktad väg från en nod a till en nod b.*

Exempel.



Figur 24.

Nät av enkelriktade gator, nätverk av oljeledningar och många andra tillämpningar kan beskrivas med en riktad graf.

Givet en graf (V, E) med $V = \{v_1, ..., v_n\}$. Låt $A = (a_{i,j})$ vara grannmatrisen till grafen. Då gäller

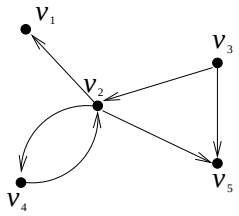
$$\text{grad}(v_i) = \sum_{j=1}^{n} a_{i,j} = \sum_{j=1}^{n} a_{j,i}\,.$$

Antag att (V, P) är en riktad graf, $V = \{v_1, ..., v_n\}$. Då är grafens grannmatris $A = (a_{i,j})$ definierad av

$$a_{i,j} = \begin{cases} 1, & \text{om det finns en pil från } v_i \text{ till } v_j; \\ 0, & \text{annars.} \end{cases}$$

Notera att nu är A i regel inte symmetrisk.

Exempel.



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

Figur 25.

Sats 5. *Låt A vara grannmatrisen till en riktad graf (V, P). Då är $(A^m)_{i,j}$ = antalet riktade vägar från v_i till v_j av längden m, där $m = 1, 2, \ldots$.*

Låt oss betrakta A i ovanstående exempel. Nu gäller

$$A^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} .$$

Exempelvis $(A^2)_{3,1} = 1$ svarar mot riktade vägen v_3, v_2, v_1 av längden 2. Vidare är $(A^2)_{3,2} = 0$, ty det finns ingen riktad väg från v_3 till v_2 av längden 2, endast av längden 1.

Bevis: (Av Sats 5, Induktion). Klart att satsen gäller för $m = 1$. Antag att satsen gäller för $A^m, m \geq 1$. Då gäller:

$$A^{m+1} = A\,A^m\,.$$

Låt $C = A^{m+1}$ och $B = A^m$, dvs. $C = A\,B$. Definitionen på matrismultiplikation ger att

$$C_{ik} = A_{i1}\,B_{1k} + \ldots + A_{ij}\,B_{jk} + \ldots + A_{in}\,B_{nk}\,. \qquad (*)$$

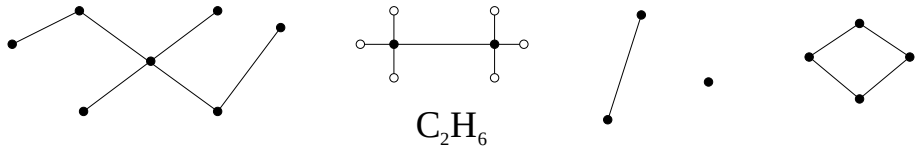
Betrakta termen $A_{ij}\,B_{jk}$. Vi vet att A_{ij} = antalet riktade vägar från v_i till v_j av längden 1. Induktionsantagandet ger att B_{jk} = antalet riktade vägar från v_j till v_k av längden m. Alltså $A_{ij}\,B_{jk}$ = antalet riktade vägar från v_i till v_k av längden $m+1$ och av formen $v_i, v_j, \ldots, v_k,\ j = 1, \ldots, n$. Då ger $(*)$ antalet riktade vägar från v_i till v_k av längden $m+1$. □

Träd (ändliga grafer)

Vi behandlar enbart ändliga grafer och definierar begreppet träd genom:

Definition. *En graf (V, E) är ett träd om den är sammanhängande och saknar cykler.*

Exempel.

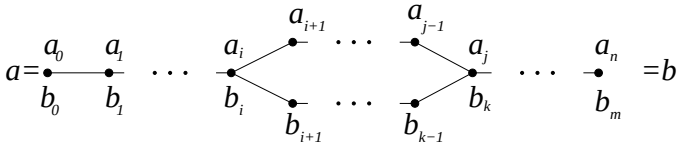


Figur 26. De två första graferna är träd medan de två andra inte är det.

En viktig egenskap hos träd är följande resultat.

Sats 6. *Låt $T = (V, E)$ vara ett träd och antag att $a, b \in V$, $a \neq b$. Då finns det exakt en väg i T från nod a till nod b.*

Bevis: Antag att $a = a_0, a_1, ..., a_n = b$ och $a = b_0, b_1, ..., b_m = b$ är olika vägar. Låt $i \geq 0$ vara det minsta index sådant att $a_{i+1} \neq b_{i+1}$, och låt $j > i + 1$ vara det minsta index sådant att det existerar ett k med $a_j = b_k$.



Figur 27.

Då är $a_i, a_{i+1}, ..., a_j, b_{k-1}, ..., b_{i+1}, a_i$ en cykel i T, vilket ger en motsägelse. Därmed finns det exakt en väg från a till b. □

Ett träd kan konstrueras genom att successivt lägga till en båge med nod till grafen.

Sats 7. *Om $T = (V, E)$ är ett träd och $|V| > 1$, så finns det åtminstone två noder av grad* 1.

Bevis: Välj en väg $a_0, a_1, ..., a_n$ av maximal längd i T. Då $|V| > 1$ och T är ett träd är $a_0 \neq a_n$. Nu måste $\text{grad}(a_0) = \text{grad}(a_n) = 1$ gälla, ty annars kunde vi fortsätta vägen som då ej vore av maximal längd. □

Sats 8. *Varje träd $T = (V, E)$ med n stycken noder är rekursivt definierat genom delträden T_i, där T_1*

innehåller en nod och T_{i+1} konstrueras genom att T_i utökas med en nod a_{i+1} och en båge mellan a_{i+1} och en av noderna i T_i.

Bevis: (Induktion). För $n = 1$ är T_1 ett träd. Antag att konstruktionen gäller för träd med upp till n-1 noder ($n \geq 2$). Tag $T = (V, E)$ med $|V| = n$. Sats 7 ger att det finns en nod $a_n \in V$: grad$(a_n) = 1$. Stryk a_n och bågen från a_n. Den graf G vi erhåller är sammanhängande och utan cykler, ty T saknar cykler. Alltså är G ett träd med n-1 noder. Induktionsantagandet ger att $G = T_{n-1}$, rekursivt konstruerad med hjälp av $T_i,\ i = 1, ..., n$-1. Sätt $T_n = T$, vilket ger den önskade följden av delträd. □

Korollarium 9. *Låt $T = (V, E)$ vara ett träd. Då är $|V| = |E| + 1$.*

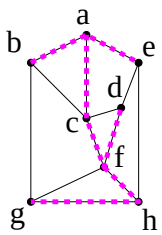
Definition. *Antag att $G = (V, E)$ är en sammanhängande graf och att T är en delgraf av G sådan att*

(i) T är ett träd;

(ii) T är en uppspännande delgraf för G.

Då säges T vara ett uppspännande träd för G.

Exempel.



Figur 28. Den "färglagda" grafen utgör ett uppspännande träd för grafen.

Det är lätt att åstadkomma ett uppspännande träd på följande sätt: Tag vilken nod som helst i en graf som "startträd" och lägg till bågar en efter en så att varje båge förenar en ny nod till startträdet (jämför Sats 8).

Det uppspännande trädet i ovanstående exempel kunde åstadkommas genom att starta med nod a och förena den med de andra noderna i ordningen b, c, e, f, d, h, g genom att lägga till bågarna $ab, ac, ae, cf, fd, fh, hg$.

Allmänt, om det finns n noder så skall vi fortsätta med n-1 steg, varefter vi har $1 + (n\text{-}1) = n$ noder och n-1 bågar. Detta är det korrekta antalet enligt Korollarium 9.

Uppspännande träd har många tillämpningar. Exempelvis, hur bygger man det billigaste vägnätet mellan ett visst antal städer? Kostnaden för vägen mellan ett par av städer beror t.ex. på hurudan terrängen är mellan städerna. Formellt har vi en graf $G = (V, E)$ vars noder är städer och vars bågar är vägarna mellan städerna, samt en funktion $w : E \to \mathbf{N}$, (där $\mathbf{N}$ betecknar de naturliga talen), sådan att $w(e)$ anger kostnaden för att bygga bågen e. Vi säger att G och w bildar en viktad graf och w är en viktfunktion.

Om man vill bygga det billigaste vägnätet mellan ett visst antal städer, så svarar vägnätet mot ett upp-

spännande träd T för G, vars totala vikt

$$w(T) = \sum_{e \in T} w(e)$$

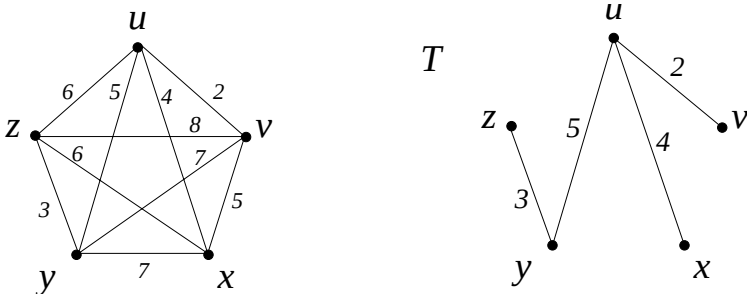
är så liten som möjligt. Detta brukar kallas MST-problemet, (minimum spanning tree problem), för den viktade grafen G.

Eftersom den viktade grafen G är ändlig har den ett ändligt antal bågar och därmed finns det ett ändligt antal uppspännande träd T för G. Det finns då, för de uppspännande träden för G, ett ändligt antal heltalsvärden $w(T)$ att välja mellan. Med andra ord finns det ett minimalt uppspännande träd T_0 för G sådant att $w(T_0) \leq w(T)$ för alla uppspännande träd T för G. Notera att det kan finnas flera uppspännande träd för G med denna egenskap.

Kruskals algoritm är en enkel algoritm som löser MST-problemet:

- (i) *Välj den kortaste (billigaste) bågen. Om det finns många bågar med samma vikt, välj en av bågarna.*
- (ii) *Om k stycken bågar har valts, välj en ny båge med minimal vikt så att ingen cykel bildas tillsammans med tidigare valda bågar.*
- (iii) *Om vi inte har ett uppspännande träd, så gå till steg (ii).*
- (iv) *Avsluta algoritmen. Det erhållna uppspännande trädet har minimal vikt.*

Exempel.



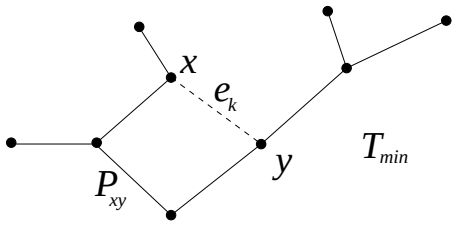
Figur 29. (i) Vi Startar med uv. (ii) + (iii) Därefter väljs zy, ux, uy. (iv) Vi erhåller ett minimalt uppspännande träd T med $\omega(T) = 14$.

Sats 10. *Låt $G = (V, E)$ vara en sammanhängande graf med viktfunktionen $w : E \to \mathbf{N}$, och antag att T är ett uppspännande träd för G som konstruerats med Kruskals algoritm. Då gäller det att*

$$w(T) \leq w(U)$$

för varje uppspännande träd U för G.

Bevis: Låt $e_1, ..., e_n$ vara bågarna i T i den ordning de erhålls ur Kruskals algoritm. Låt $T_{\min}$ vara ett minimalt uppspännande träd för G. Om $T_{\min} = T$ så gäller satsens påstående. Om $T_{\min} \neq T$ betecknar vi med e_k den första bågen för T i Kruskals algoritm som inte är en båge i $T_{\min}$. Låt $e_k = \{x, y\}$ och låt P_{xy} vara vägen i $T_{\min}$ som förenar noderna x och y.



Figur 30.

Om bågen e_k läggs till $T_{\min}$, så har grafen $T_{\min} \cup \{e_k\}$ en cykel $c = e_k \cup P_{xy}$ och då T inte har någon cykel, måste c innehålla åtminstone en båge e'_k som inte finns i T. Avlägsna denna båge och erhåll trädet

$$T'_{\min} = (T_{\min} \cup \{e_k\}) \setminus \{e'_k\},$$

med samma noder som $T_{\min}$ och med totala vikten

$$w(T'_{\min}) = w(T_{\min})\text{-}w(e'_k) + w(e_k)\,.$$

Då $T_{\min}$ är ett minimalt uppspännande träd måste det gälla att

$$w(e'_k) \leq w(e_k)\,. \qquad (*)$$

Men e_k är bågen med minimal vikt, sådan att ingen cykel bildas då e_k läggs till $e_1, ..., e_{k-1}$, (Kruskals algoritm).

Emedan $e_1, ..., e_{k-1}, e'_k$ är bågar i $T_{\min}$ får vi ingen cykel då e'_k läggs till $e_1, ..., e_{k-1}$. Då gäller det att $w(e'_k) \geq w(e_k)$ och med stöd av $(*)$ att

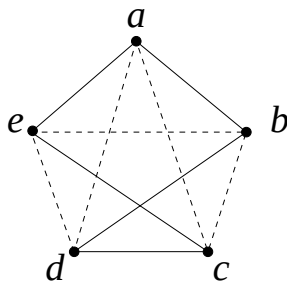
$$w(e'_k) = w(e_k),$$

varför $w(T'_{\min}) = w(T_{\min}) =$ minimal total vikt. Vi har därmed hittat ett minimalt uppspännande träd $T'_{\min}$ med en båge mera än $T_{\min}$ gemensam med T, (bågen e_k). Upprepning av ovanbeskrivna procedur ger slutligen ett minimalt uppspännande träd som sammanfaller med T, dvs. $w(T_{\min}) = w(T'_{\min}) = ... = w(T)$. □

Exempel. Handelsresandesproblem. En handelsresande önskar besöka ett antal städer och sedan återvända hem med minsta möjliga totala kostnad utan att återvända till en stad han redan besökt. Den handelsresande bör alltså hitta en Hamilton-cykel med minimal vikt. En lösning behöver inte alltid existera, men om en lösning finns så är antalet cykler i allmänhet stort och det är ett övermäktigt kombinatoriskt problem att gå igenom alla alternativ.

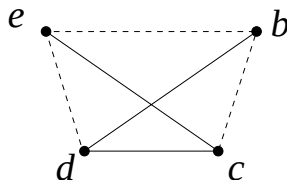
Vi skall här beskriva en systematisk metod som med hjälp av Kruskals algoritm bestämmer en undre gräns för den eventuella lösningen till handelsresandesproblem.

Antag att cykeln i grafen nedan ger en lösning till handelsresandesproblem för städerna a, b, c, d, e.



Figur 31.

Om vi tar bort nod a, så får vi en väg genom noderna b, d, c, e.



Figur 32.

Denna delgraf bildar ett uppspännande träd U för den fulla delgrafen med noderna b, c, d, e. Låt T vara det uppspännande träd som kan konstrueras med Kruskals algoritm. Då ger Sats 10 att

$$w(U) \geq w(T)\,.$$

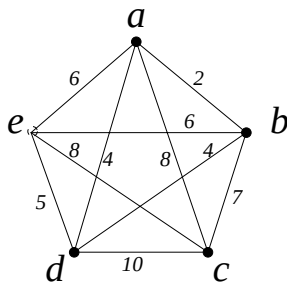
Hela cykeln har således den totala vikten

$$w(U) + w(\{e, a\}) + w(\{a, b\}) \geq w(T) + w(\{e, a\}) + w(\{a, b\})\,.$$

Med andra ord kan vi uppnå en undre gräns för lösningen till handelsresandesproblem genom att addera den totala vikten av det minimala uppspännande träd som konstruerats med Kruskals algoritm genom noderna b, c, d och e med vikterna för de två "minsta" bågarna utgående från a.

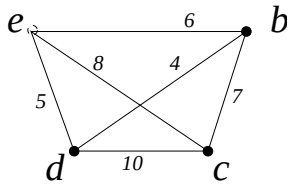
Följande exempel visar hur vi skall använda denna metod.

Exempel. Betrakta fem städer med vikterna enligt följande figur:



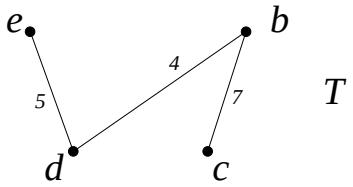
Figur 33.

Om vi avlägsnar nod a, så får vi följande graf med noderna b, c, d och e.



Figur 34.

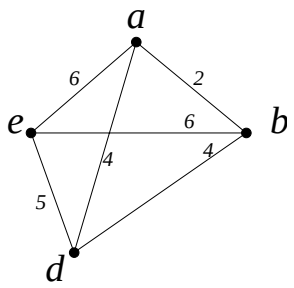
Kruskals algoritm ger då ett uppspännande träd med bågarna bd, de, bc:



Figur 35.

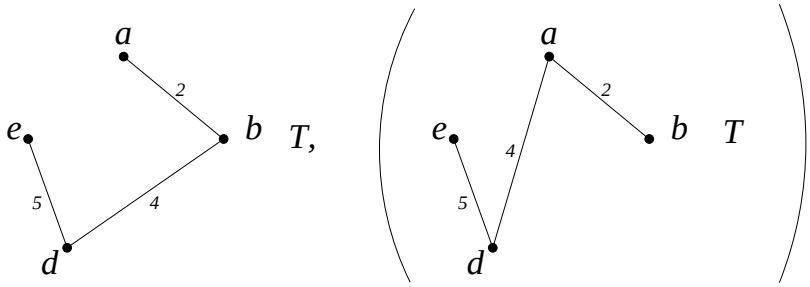
med totala vikten $w(T) = 16$. De två minsta vikterna för bågar utgående från a är 2 och 4 (bågarna ab och ad). Alltså den undre gränsen för lösningen till handelsresandesproblem är i detta fall, då vi avlägsnade noden a, given av $16 + 2 + 4 = 22$.

Låt oss nu undersöka vad som inträffar då vi avlägsnar nod c. Då får vi följande graf med noderna a, b, d och e.



Figur 36.

Tillämpning av Kruskals algoritm ger det uppspännande trädet T med bågarna ab, bd, de (eller ab, ad, de), alltså:



Figur 37.

med totala vikten $w(T) = 11$. De två minsta vikterna för bågar utgående från c är 7 och 8 (bågarna cb och ca eller ce). Därmed får vi den undre gränsen $11 + 7 + 8 = 26$ till handelsresandesproblem, vilket ger en bättre uppskattning än i det föregående fallet där noden a avlägsnades.

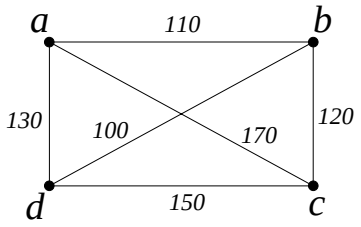
Alltså är lösningen till handelsresandesproblem åtminstone 26. Genom att avlägsna noderna b, d och e i tur och ordning får man undre gränser som är mindre än 26 (hemuppgift), alltså sämre undre gränser.

Det visar sig att den undre gräns vi fick genom att avlägsna noden c råkar ge en lösning till handelsresandesproblem, nämligen cykeln med minimal total vikt genom noderna a, b, c, d, e, är $abceda$ med den totala vikten $2 + 7 + 8 + 5 + 4 = 26$.

I nästa exempel presenteras två "giriga algoritmer" som i en fullständig viktad graf snabbt hittar en Hamilton-cykel, vilken dock inte behöver vara lösningen till handelsresandesproblem.

Exempel. Två algoritmer som ger en "ganska kort" Hamilton-cykel i fullständiga grafer.

En graf (V, E) med $V = \{v_1, ..., v_n\}$ är fullständig, (eller komplett), om grad$(v_i) = n$-1, $i = 1, ..., n$. Betrakta grafen:



Figur 38.

A. Metoden med närmaste granne

(i) Starta i någon nod x. Välj utgående båge med minimal vikt.
(ii) Låt y vara den senast uppnådda noden. Om alla noder i V är inkluderade i vägen så väljs bågen mellan y och x. Välj annars en båge med minimal vikt utgående från y till en nod som inte ännu inkluderats i vägen.

I vår exempelgraf erhålls

1. $x = a:\ a, b, d, c, a \ \Rightarrow 110 + 100 + 150 + 170 = 530,$
2. $x = b:\ b, d, a, c, b \ \Rightarrow 100 + 130 + 170 + 120 = 520,$
3. $x = c:\ c, b, d, a, c \ \Rightarrow 120 + 100 + 130 + 170 = 520,$
4. $x = d:\ d, b, a, c, d \ \Rightarrow 100 + 110 + 170 + 150 = 530.$

B. Metoden med sorterade bågar

(i) Välj den kortaste bågen bland alla bågar som ännu inte är valda, så att aldrig fler än två bågar utgår från en nod och så att inga cykler av längd $< n$ bildas.

I vårt exempel väljs bågarna i ordningen $bd, ab, cd,$ och ac. Vi får Hamilton-cykeln a, b, d, c, a med längden 530.

Algoritmerna A och B är exempel på giriga algoritmer (greedy algorithms) som är snabba, men som inte nödvändigtvis hittar den optimala lösningen. Lösningen till handelsresandesproblem i vårt fall ges av Hamilton-cykeln a, b, c, d, a med längden 510.

I vårt exempel med fem städer lyckas båda algoritmerna lösa handelsresandesproblem. (Kolla!)

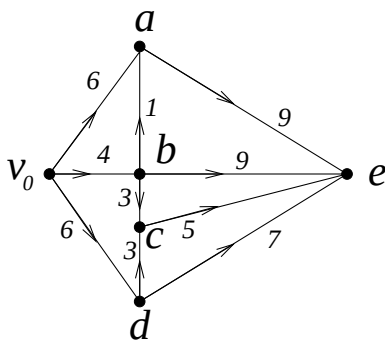
Kortaste vägen i riktade grafer

Antag att vi har en riktad graf $G = (V, P)$ där vi förser varje pil $p \in P$ med en vikt $w(p)$ med hjälp av viktfunktionen $w : P \to \mathbf{N}$. (De tidigare behandlade viktade graferna kan betraktas som specialfall).

Låt nu $v_0, v_n \in V$ vara två noder i G sådana att det existerar åtminstone en riktad väg från v_0 till v_n. Vi skall nu med hjälp av ett exempel beskriva Dijkstras algoritm från 1959 för bestämning av kortaste vägen mellan v_0 och v_n. (Edsger W. Dijkstra, 1930-2002). I algoritmen kan en nod v_j tilldelas två typer av

markeringar, en temporär markering (n, v_k), som kan ändras, eller en permanent markering $[n, v_i]$ som inte mera ändras. I dessa markeringar anger n längden på den för närvarande respektive den slutliga kortaste vägen från v_0 till v_j, där den sista pilen börjar i v_k respektive v_i och slutar i v_j. Vi är därmed intresserade av att bestämma den permanenta markeringen för noden v_n.

Exempel. Bestäm den kortaste vägen från nod v_0 till nod e i nedanstående viktade riktade graf:



Figur 39.

Steg 1: Vi förser startnoden v_0 med den permanenta markeringen $v_0 : [0, -]$.
Steg 2: Noder som nås direkt från $v_0 : a, b, d$. Dessa förses med de temporära markeringarna

$$a : (6, v_0) \ , \ b : (4, v_0) \ , \ d : (6, v_0) \, .$$

Steg 3: Den minsta temporära markeringen görs permanent

$$b : [4, v_0] \, .$$

Steg 4: Noder som nås direkt från $b : a, c, e$. Nod a har redan tilldelats en temporär markering, men eftersom vägen från v_0 via b till a är kortare, $4 + 1 = 5 < 6$, så ändras markeringen på nod a,

$$a : (5, b) \ , \ c : (7, b) \ , \ e : (13, b) \, .$$

Steg 5: Den minsta temporära markeringen görs permanent

$$a : [5, b] \, .$$

Steg 6: Noder som nås direkt från $a : e$. Eftersom $5 + 9 = 14 > 13$ ändrar vi inte på den temporära markeringen för nod e.
Steg 7: Den minsta temporära markeringen görs permanent

$$d : [6, v_0] \, .$$

Steg 8: Noder som nås direkt från $d : c, e$. Vi har att $6 + 3 > 7$ och att $6 + 7 = 13$, så vi ändrar inga temporära markeringar.

Steg 9: Minsta temporära markeringen görs permanent

$$c : [7, b] .$$

Steg 10: Endast noden e nås direkt från nod c. Noterar att $7 + 5 < 13$, så vi ändrar den temporära markeringen för nod e

$$e : (12, c) .$$

Steg 11: Den enda kvarvarande temporära markeringen görs permanent

$$e : [12, c] .$$

Nu kan vi på basen av de permanenta markeringarna göra bakåtanalysen: $e \leftarrow c \leftarrow b \leftarrow v_0$. Därmed ges den kortaste vägen av v_0, b, c, e och den har längden 12.