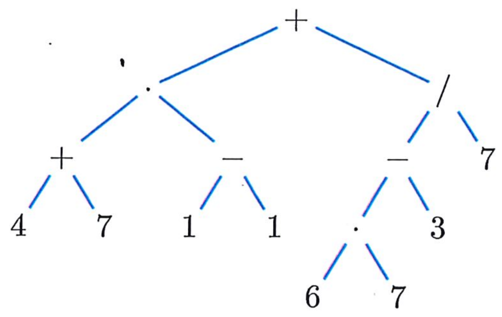
(c)



Övning 6.87 Konstruera träd för följande två uttryck givna i postordning:

(a) $6\ 2\ 3 \cdot -\ 5\ 1 - +$ (b) $4\ 3 + 2\ 9 - +\ 600\ 6\ / \cdot$

(c) Beräkna värdet av de två uttrycken ovan!

Övning 6.88: *Viktig!* *Plus* är ett exempel på en **binär operation** (*eng.* binary operation), det vill säga en operation som verkar på två tal (såsom $3 + 5$). Binära operationer är väldigt vanliga men det finns även operationer som bara verkar på ett enda tal, exempelvis *roten ur*: $\sqrt{3}$. Sådana operationer kallas **unära**.

(a) Vilken vanlig symbol betecknar både en binär operation och en unär operation?

(b) Hur kan man med binära träd modellera uttryck med även unära operationer?

6.6 Tre exempel på modellering med grafer

Som vi redan har sett finns det många situationer som självklart modelleras med grafer, såsom vägnät och sociala nätverk. Men grafer kan vara användbara modeller även i mindre uppenbara sammanhang. I detta avslutande avsnitt ger vi tre exempel på detta:

- *Tidsplanering* i till exempel byggprojekt.
- *Radbrytning* i ordbehandlingsprogram.
- Lösning av *Instant Insanity*, ett välkänt knep-och-knåp.

Dessa problem förefaller vid första anblicken inte ha särskilt mycket med grafer att göra, men kan vid djupare analys hanteras med grafteoretiska metoder.

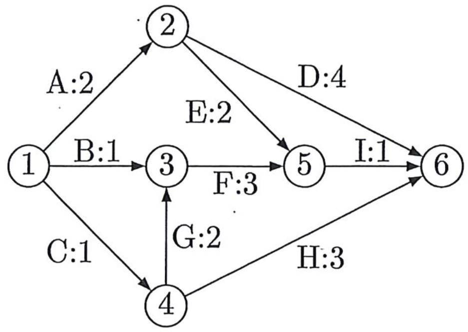
Kom ihåg att kärnpunkten när något ska modelleras med grafer är att välja ut vad som ska representeras av *noder* och vad som ska representeras av *kanter*.

6.6.1 Tidsplanering

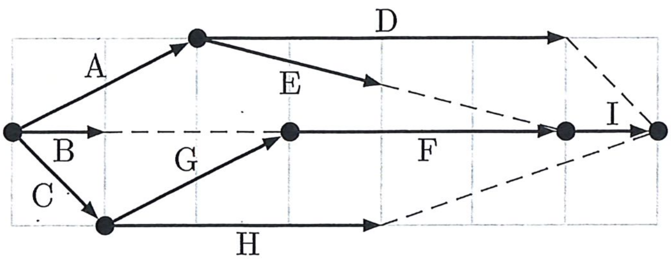
Att göra fungerande tidsplaner för byggen har alltid varit problematiskt. Ett byggprojekt brukar bestå av ett antal separata arbetsuppgifter. Det går hyfsat att bedöma hur lång tid varje enskild arbetsuppgift kommer att ta – men detta säger ingenting om helheten! Problemet är att en del arbeten kan utföras parallellt (man kan exempelvis måla innertaken samtidigt som man putsar fasaden) medan andra uppgifter måste komma i en viss ordning (man kan inte måla innertaket innan man byggt det).

Den här problemställningen är mycket lämplig att modellera med hjälp av grafer. De i tiden utsträckta arbetsuppgifterna får bli kanter, och tidpunkterna då de påbörjas eller avslutas får bli noder.

Här är en graf som visar ett mindre projekt. Märkningen på kanterna är momentets namn samt det antal veckor man uppskattar att momentet kommer att ta.



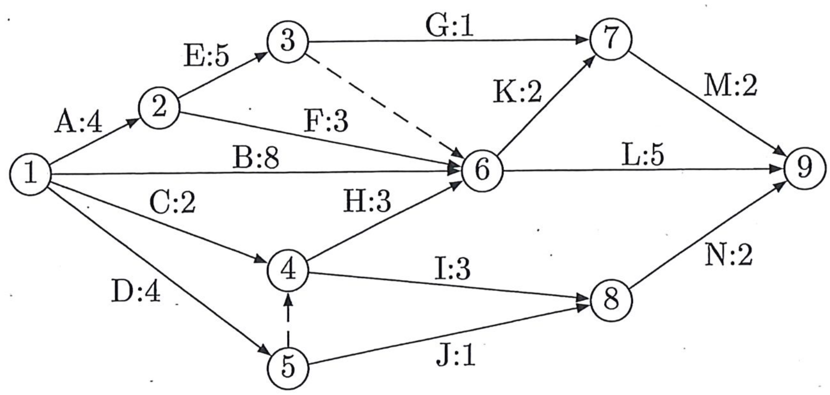
Vill vi nu veta hur lång tid projektet kommer att ta kan vi rita upp det hela på ett rutat papper, med tidsskala i horisontalled. En ruta får motsvara en vecka. En del pilar kommer då att bli utdragna över större tidsrymd än vad arbetet faktiskt tar. Det markerar vi genom att markera själva arbetet med en tjock pil och eventuell väntetid i slutet med en streckad linje. Just det här projektet får följande tidsgraf:



Ur grafen kan vi exempelvis utläsa att hela projektet bör ta 7 veckor; det är arbetsmomenten C, G, F och I – som måste utföras i precis den ordningen – som bestämmer tiden, medan en försening i övriga moment troligen inte gör något. (Man säger att momenten C, G, F och I är *tidskritiska*, vilket innebär att en försening där försenar hela bygget.)

Den här beskrivna metoden att tidsplanera lär ha uppfunnits i samband med att man sökte något sätt att datorisera tidsberäkningarna, och visade sig så pass bra att det (i alla fall vid mindre projekt) var överflödigt att blanda in datorer i beräkningen!

Övning 6.89 Rita en tidsgraf för nedanstående bygge. (Den streckade pilen mellan punkt 3 och punkt 6 anger att tidpunkt 6 inte får komma före tidpunkt 3, då de arbeten som ska starta i tidpunkt 6 fordrar att arbetsuppgift E är klar. Samma resonemang för den streckade pilen mellan tidpunkt 5 och tidpunkt 4.) Tala också om hur lång tid bygget kommer att ta och vilka arbetsmoment som är tidskritiska. (Angivna tider är i dagar.)



(Uppgiften är för övrigt saxad ur en skrivning i byggnadsproduktion.)

6.6.2 Radbrytning i TEX

Ett stycke i en text är ju egentligen en lång, sammanhängande enhet, som delats upp i rader på pappret, eftersom man annars skulle behöva boksidor som var ett par meter breda. I de flesta böcker har man sedan manipulerat raderna så att de alla blir lika långa. Tidigare har detta varit något som sättarna på tryckerierna gjort efter eget omdöme, men i och med de datoriserade sättsystemens intåg har det blivit nödvändigt att automatisera processen. Det finns många olika sätt att göra detta. Det här beskrivna uppfanns av den amerikanske professorn Donald E. Knuth, som blev förbannad då han såg hur ful hans senaste bok hade blivit. (Det är för övrigt hans program, TEX, som vi använt till den här boken.)

Då man ska anpassa en rad i texten så att den ska få rätt längd får man utnyttja mellanrummen mellan orden. De kan vid behov tryckas ihop lite grand, eller töjas ut. Man kan inte trycka ihop dem hur mycket som helst, men det finns inga gränser för hur mycket man kan töja dem, frånsett att det ser väldigt dumt ut. Vidare har man möjlighet att bryta av ord som bara delvis ryms på raden, genom att avstava dem. Det minskar läsbarheten en aning, men kan vara bättre än att få flera centimeter stora ordmellanrum.

Då vi ska bryta upp vårt stycke kan vi överväga flera alternativa sätt att göra det. I det här stycket skulle vi kunna avsluta första raden efter "att" eller efter "göra" eller kanske avstava "göra". Sedan gör vi samma överväganden för nästa rad. Vilka möjliga brytpunkter den har beror nu delvis på var vi bröt den första raden; om vi brutit första raden i det här stycket efter "att" skulle det inte gå att bryta andra efter "eller", medan det är möjligt då vi brutit den första efter "göra".

Alla de tänkbara sätten att bryta stycket kan modelleras med en riktad graf. Som noder har vi de tänkbara brytpunkterna. Två punkter är förbundna med en kant om det är tänkbart att sätta en rad som börjar i ena punkten och slutar i andra. En brytning av stycket motsvaras av en stig från den nod som representerar styckets början till den nod som representerar dess slut.

Det vi nu vill göra är att välja den stig som motsvarar det vackraste (eller minst fula) alternativet. Det gör vi genom att tilldela varje tänkbar rad ett fulhetsvärde. En rad med alldeles lagom stora ordmellanrum och ingen avstavning har fulhetsvärdet 0. Ju mer ordmellanrummen avviker från idealet, ju högre fulhetsvärde får raden. En avstavning ökar också fulhetsvärdet. Vackraste alternativet blir då den stig som har minst sammanlagt fulhetsvärde. Vi har i praktiken skrivit om problemet att hitta ett snyggt sätt att bryta stycket till problemet att hitta kortaste vägen mellan två punkter. (Hur man sedan löser *det* problemet ligger utom ramen för den här boken, men tas upp i nästa del.)

Övning 6.90 Om du tittar igenom några sidor av boken kommer du att finna att det är ganska vanligt att just första raden i ett stycke avslutas med ett avstavat ord. Hur kommer det sig?

Övning 6.91 De flesta ordbehandlare och sättsystem använder inte så här sofistikerade metoder. Undersök något program som du har tillgång till, och försök analysera hur just det programmet beslutar var radbrytningarna ska sitta.

6.6.3 Instant Insanity

Instant Insanity är ett knep-och-knåp bestående av fyra kuber. Var och en av de sex sidorna på varje kub är målad i någon av fyra färger (låt oss säga blått–B, svart–S, grått–G eller vitt–V). Målsättningen är att bygga ett torn av klossarna, så att samtliga färger finns med på samtliga sidor av tornet.

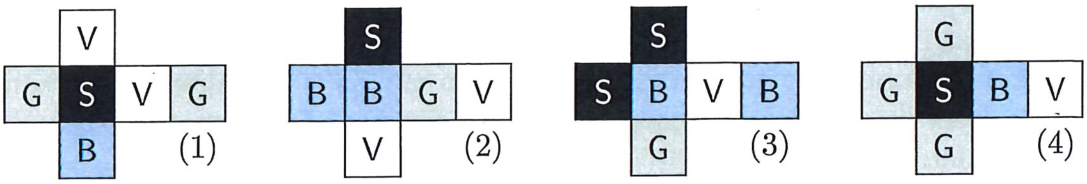
Det finns 41 472 olika torn man kan bygga med fyra olika klossar (se övning 6.95 på sidan 169). Det är alltså inte praktiskt genomförbart att helt enkelt bygga alla torn som går, och se efter om något av dem uppfyller kraven.

Om vi börjar med att försöka göra framsidan på tornet snygg så finner vi att vi då också utformar baksidan, eftersom klossarnas motstående sidor kommer att synas där. Det verkar därför klokt att på något sätt försöka att *samtidigt* lägga färgerna på framsidan och baksidan. För detta behöver vi bokföra vilka färger som sitter på motstående sidor på klossarna. Vi kan göra detta genom att rita en graf (en multigraf) med en nod för varje färg, och en kant mellan två färger om de befinner sig på motstående sidor av en kloss. Dessutom markerar vi på kanten vilken av klossarna den beskriver. (Se bilden i exemplet, om förklaringen var svår att förstå.)

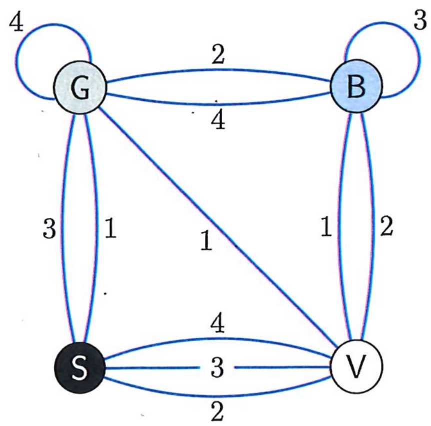
Vi vill nu vrida klossarna så att alla färger finns med både på fram- och på baksidan av tornet. Om vi plockar en av kanterna som hör till kloss 1, så står den för att kloss 1 är vriden så att färgen i ena ändan av kanten finns på framsidan av tornet och färgen i andra ändan på baksidan. Om vi plockar 4 kanter, en för varje kloss, och gör detta så att varje färg kommer med två gånger så har vi fått ett recept på hur klossarna kan vridas så att varje färg kommer med två gånger; en gång på framsidan och en på baksidan.

Om vi sedan kan plocka ytterligare 4 förbindelser enligt samma princip, har vi fått ett tilläggsrecept på hur vi ska ordna så att höger- och vänstersidan av tornet också blir mångfärgade. Och har vi ordnat det är tornet klart!

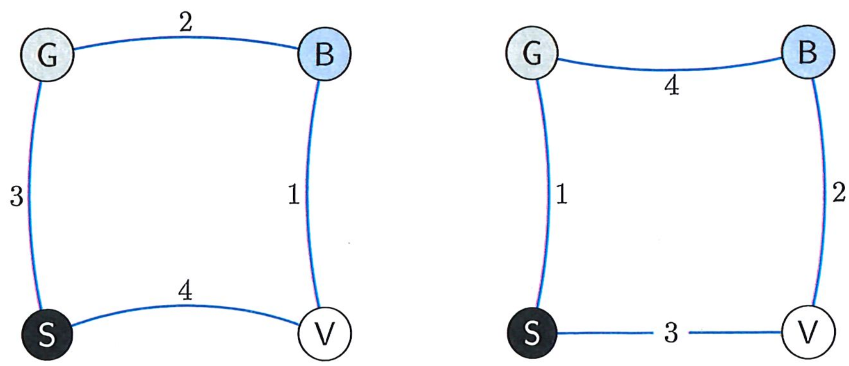
Exempel 6.6: ***Instant insanity*** Vi har nedanstående klossar (ni får tänka er att ni klipper ut dem och viker ihop dem)



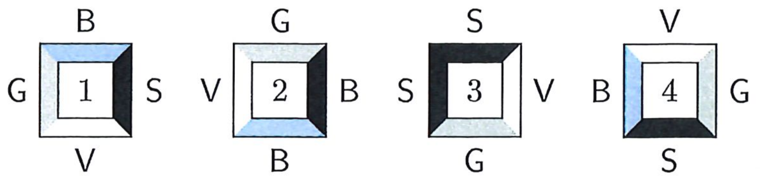
Vi ritar upp grafen över hur färgerna på motstående sidor förhåller sig till varandra



Vi försöker nu plocka ut fyra kanter ur grafen, en för varje kloss, så att varje färg är kopplad till två av kanterna. Då har vi ordnat framsidan och baksidan. Sedan försöker vi hitta fyra andra kanter enligt samma princip, för att ordna höger- och vänstersidan. Det är möjligt, enligt

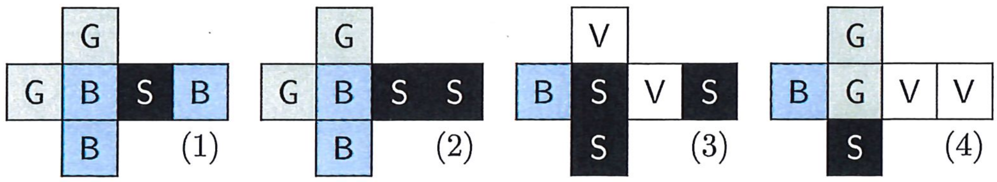


Vi kan bygga ett torn, där de fyra klossarna ligger enligt



Övning 6.92 Verifiera dels att den uppritade stora grafen verkligen motsvarar de givna klossarna, dels att det föreslagna tornet verkligen motsvaras av delgraferna, dels att tornet är möjligt att bygga med de givna klossarna, och att det uppfyller kraven. *

Övning 6.93 Här är en annan uppsättning klossar.



Hitta en lösning för dem.

Övning 6.94 Ta och måla en uppsättning klossar enligt anvisningarna i föregående övning. Prova att bygga ett torn enligt din lösning. Ge sedan klossarna till någon annan, beskriv problemet, och ta tid på hur länge det dröjer innan vederbörande i vredesmod slänger klossarna i väggen. *

Övning 6.95 Förklara varför det finns $4! \cdot 6 \cdot (6 \cdot 4)^3 = 1\,990\,656$ sätt att stapla de fyra klossarna i Instant Insanity. Förklara också varför detta antal reduceras till $1\,990\,656/(4! \cdot 2) = 41\,472$ om man inte bryr sig om i vilken ordning klossarna ligger, eller om man vänder hela stapeln upp-och-ner.

Övning 6.96 Generellt angående modellering med grafer: Vilka saker har modellerats med hjälp av grafer i tidigare kapitel? *