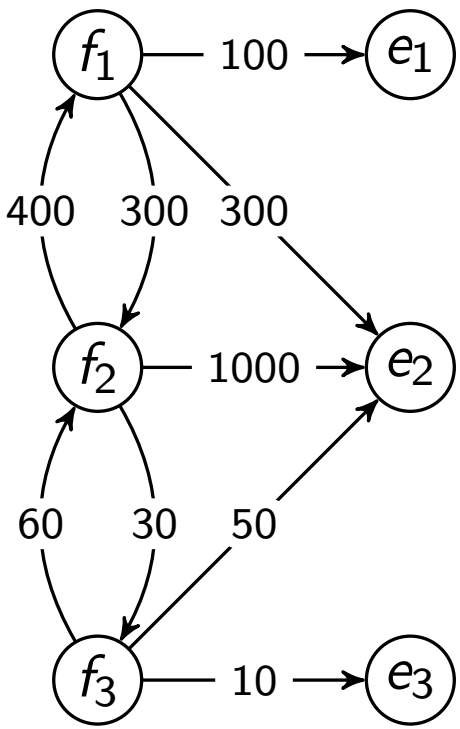
Øvingsforelesning 13

TDT4120 - Algoritmer og datastrukturer

Øving 12

Julenissens logistikk

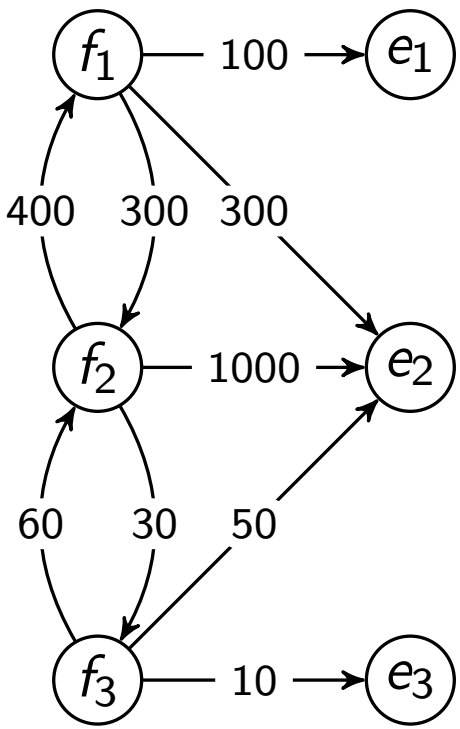
Oppgave 2: Gitt et sett med fabrikker med produksjonskapasitet, et sett med lokasjoner og muligheten til å transportere opptil et sett med gjenstander mellom disse. Hva må vi gjøre med følgende graf for å finne maksimalt antall pakker som kan produseres og sendes til alle lokasjonene som et maks-flyt-problem?



- Splitte de antiparallele kantene.
- Fjerne syklene mellom fabrikkene.
- Legge til et supersluk med kanter fra lokasjonene og en superkilde bak alle fabrikkene.
- Splitte hver fabrikk-node i to.

Julenissens logistikk

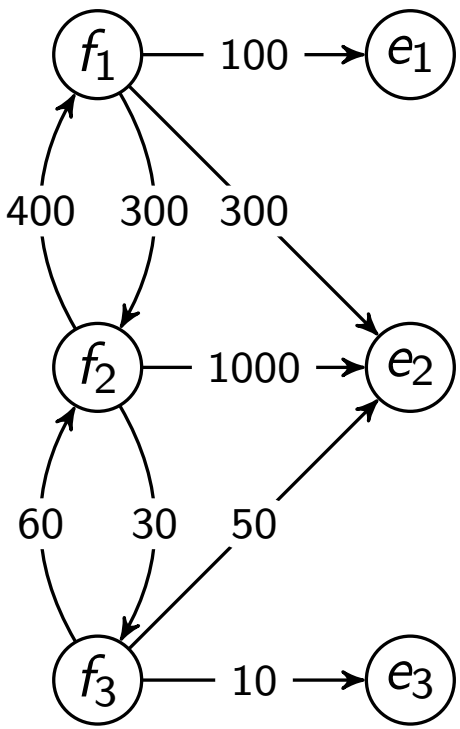
Oppgave 2: Gitt et sett med fabrikker med produksjonskapasitet, et sett med lokasjoner og muligheten til å transportere opptil et sett med gjenstander mellom disse. Hva må vi gjøre med følgende graf for å finne maksimalt antall pakker som kan produseres og sendes til alle lokasjonene som et maks-flyt-problem?



- Splitte de antiparallele kantene.
- ~~Fjerne syklene mellom fabrikkene.~~
- Legge til et supersluk med kanter fra lokasjonene og en superkilde bak alle fabrikkene.
- Splitte hver fabrikk-node i to.

Julenissens logistikk

Oppgave 2: Gitt et sett med fabrikker med produksjonskapasitet, et sett med lokasjoner og muligheten til å transportere opptil et sett med gjenstander mellom disse. Hva må vi gjøre med følgende graf for å finne maksimalt antall pakker som kan produseres og sendes til alle lokasjonene som et maks-flyt-problem?



- Splitte de antiparallele kantene.
- ~~Fjerne syklene mellom fabrikkene.~~
- Legge til et supersluk med kanter fra lokasjonene og en superkilde bak alle fabrikkene.
- ~~Splitte hver fabrikk-node i to.~~

Maksimal flyt - Definisjoner

Oppgave 3: Hva betyr notasjonen $3/6$ på en rettet kant mellom to noder u og v i et flytnett?

Maksimal flyt - Definisjoner

Oppgave 3: Hva betyr notasjonen 3/6 på en rettet kant mellom to noder u og v i et flytnett?

Flyten over kanten, $f(u, v)$ er 3

Kapasiteten til kanten, $c(u, v)$, er 6.

Maksimal flyt - Definisjoner

Oppgave 3: Hva betyr notasjonen 3/6 på en rettet kant mellom to noder u og v i et flytnett?

Flyten over kanten, $f(u, v)$ er 3

Kapasiteten til kanten, $c(u, v)$, er 6.

Oppgave 4: Gitt kanten fra oppgave 3, hva er restkapasiteten, $c_f(u, v)$?

Maksimal flyt - Definisjoner

Oppgave 3: Hva betyr notasjonen 3/6 på en rettet kant mellom to noder u og v i et flytnett?

Flyten over kanten, $f(u, v)$ er 3

Kapasiteten til kanten, $c(u, v)$, er 6.

Oppgave 4: Gitt kanten fra oppgave 3, hva er restkapasiteten, $c_f(u, v)$?

$$c_f(u, v) = c(u, v) - f(u, v) = 6 - 3 = 3$$

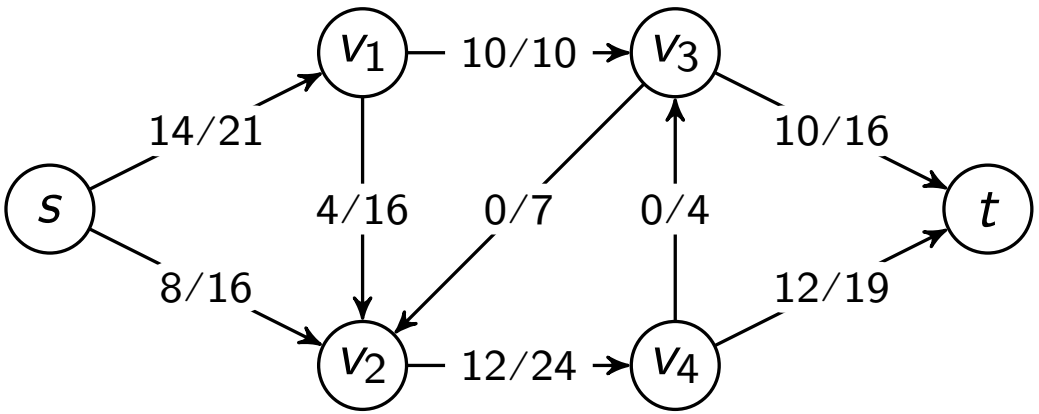
Maksimal flyt - Anvendelser

Oppgave 5: Hva har vi dersom vi løser maks-flyt-problemet fra en vilkårlig node u til en annen vilkårlig node v i en rettet, sammenhengende graf hvor alle kanter har kapasitet 1?

- Korteste vei fra u til v.
- Det maksimale antallet stier mellom u og v.
- Det maksimale antallet stier mellom u og v som ikke deler kanter.
- Det maksimale antallet stier mellom v og u som ikke deler kanter.

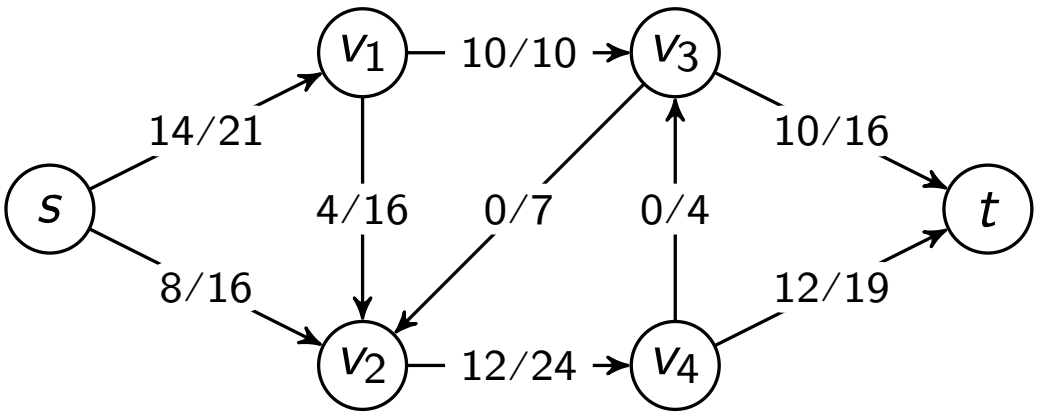
Maksimal flyt - Flytnett

Oppgave 6: Hva er flyten i dette flytnettet?



Maksimal flyt - Flytnett

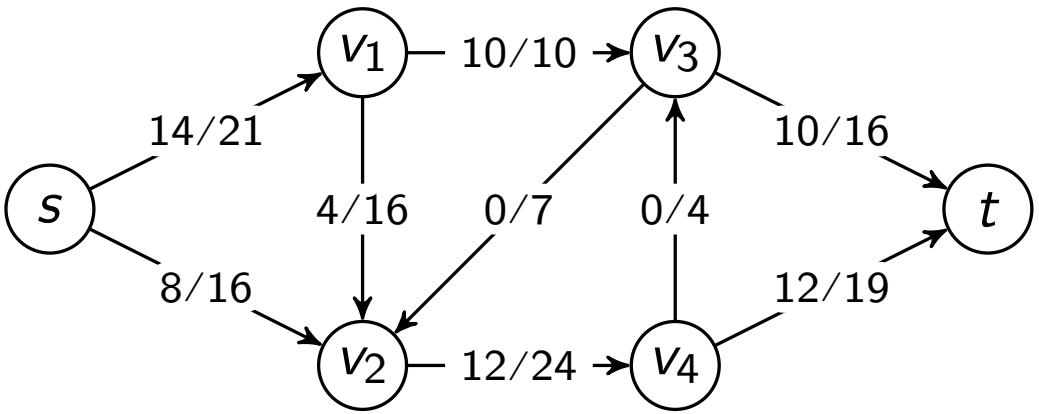
Oppgave 6: Hva er flyten i dette flytnettet?



Oppgave 7: Finn en førøkende sti i flytnettet over, ved hjelp av BFS. Hva blir den forøkende stien?

Maksimal flyt - Flytnett

Oppgave 6: Hva er flyten i dette flytnettet?

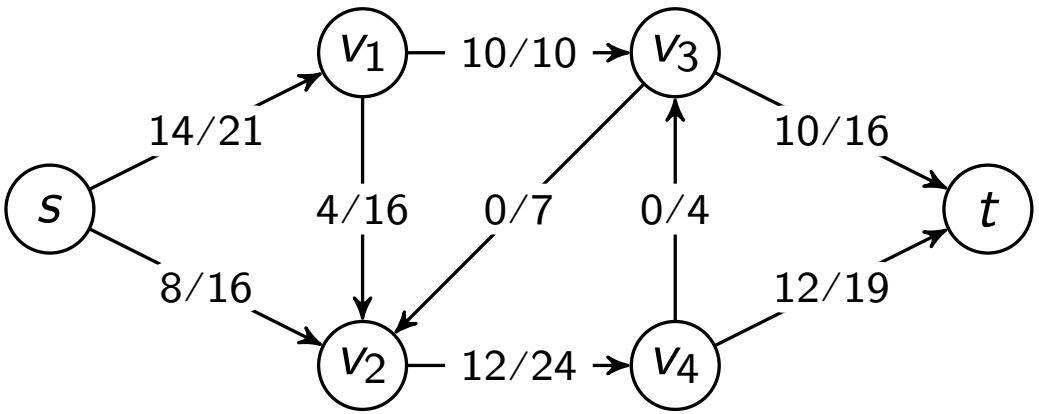


Oppgave 7: Finn en førøkende sti i flytnettet over, ved hjelp av BFS. Hva blir den forøkende stien?

Stien blir $s \to v_2 \to v_4 \to t$.

Maksimal flyt - Flytnett

Oppgave 6: Hva er flyten i dette flytnettet?



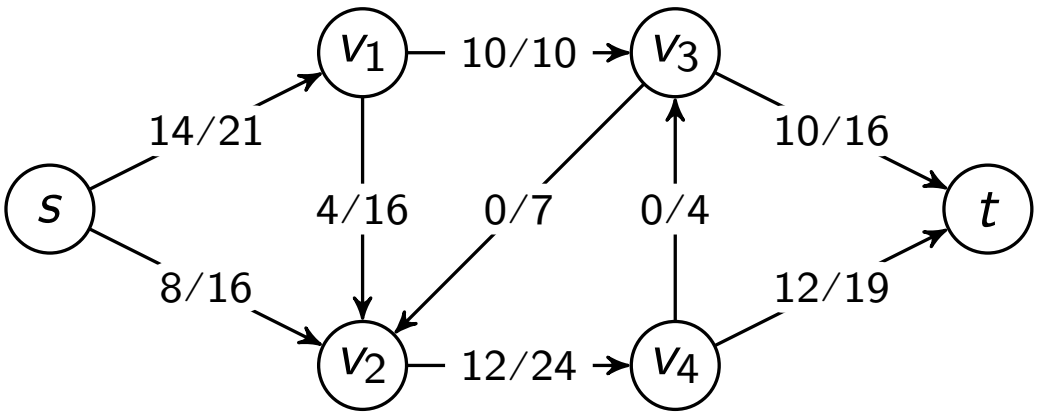
Oppgave 7: Finn en førøkende sti i flytnettet over, ved hjelp av BFS. Hva blir den forøkende stien?

Stien blir $s \to v_2 \to v_4 \to t$.

Oppgave 8: Hvor mye flyt kan sendes gjennom denne forøkende stien?

Maksimal flyt - Flytnett

Oppgave 6: Hva er flyten i dette flytnettet?



Oppgave 7: Finn en førøkende sti i flytnettet over, ved hjelp av BFS. Hva blir den forøkende stien?

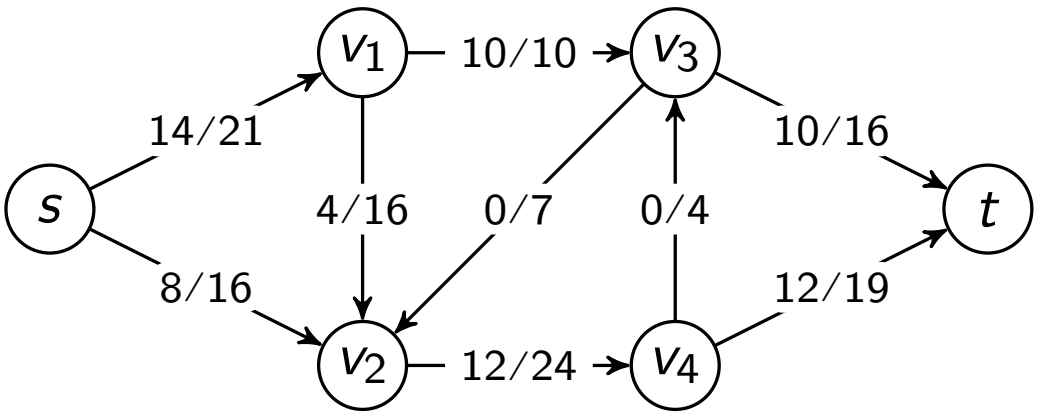
Stien blir $s \to v_2 \to v_4 \to t$.

Oppgave 8: Hvor mye flyt kan sendes gjennom denne forøkende stien?

$$\min(c_f(s, v_2), c_f(v_2, v_4), c_f(v_4, t)) = \min(8, 12, 7) = 7$$

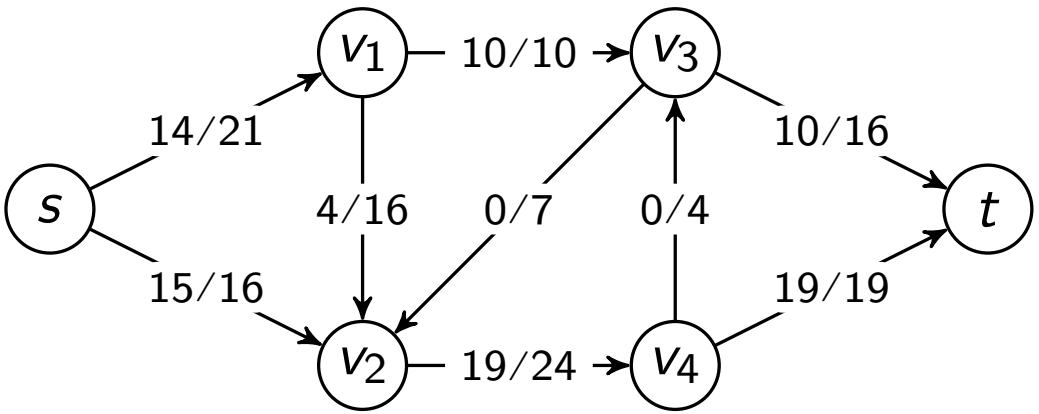
Maksimal flyt - Flytnett

Oppgave 9: Hva er den maksimale flyten i flytnettet?



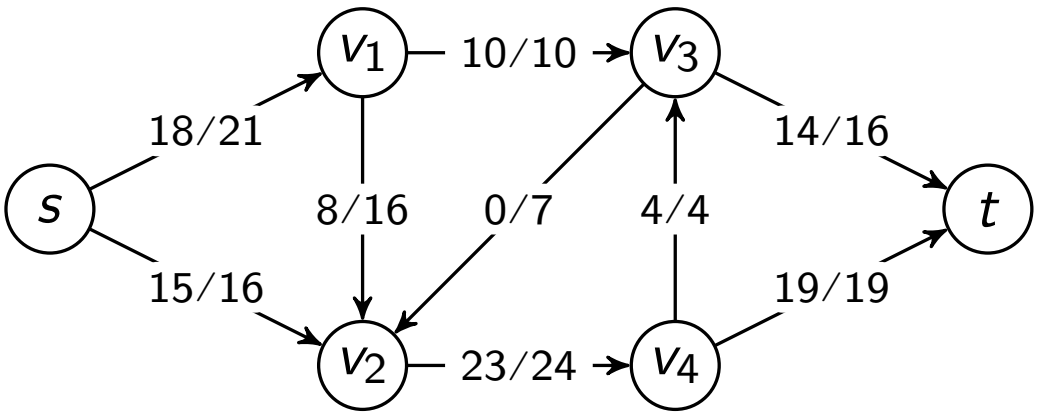
Maksimal flyt - Flytnett

Oppgave 9: Hva er den maksimale flyten i flytnettet?



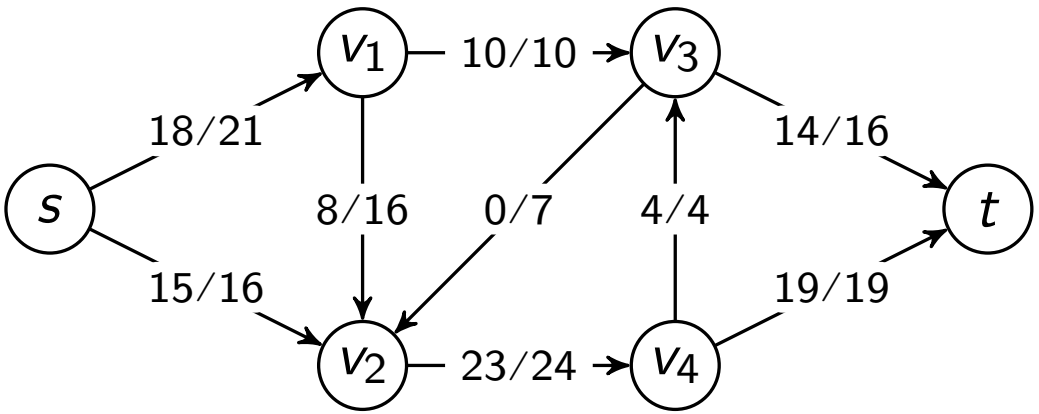
Maksimal flyt - Flytnett

Oppgave 9: Hva er den maksimale flyten i flytnettet?



Maksimal flyt - Flytnett

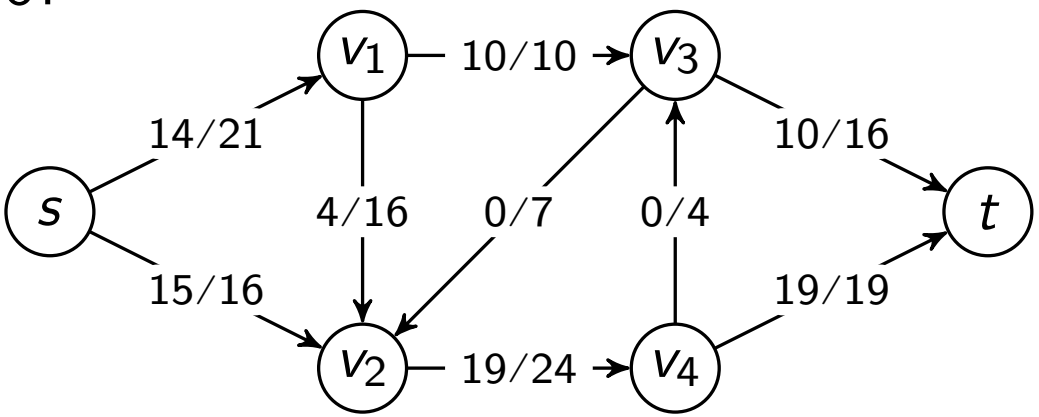
Oppgave 9: Hva er den maksimale flyten i flytnettet?



Ender opp med å finne et minimalt snitt $(\{s, v_1, v_2, v_4\}, \{v_3, t\})$ med kapasitet 33.

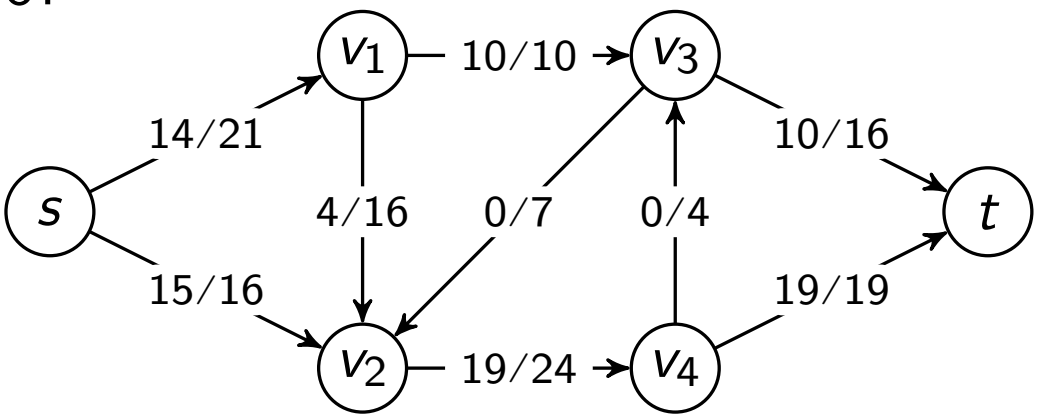
Maksimal flyt - Flytnett

Oppgave 10: Hva er flyten over snittet $(\{s, v_2\}, \{v_1, v_3, v_4, t\})$ i flytnettverket etter å ha lagt til den flytforøkende fra oppgave 7 og 8?



Maksimal flyt - Flytnett

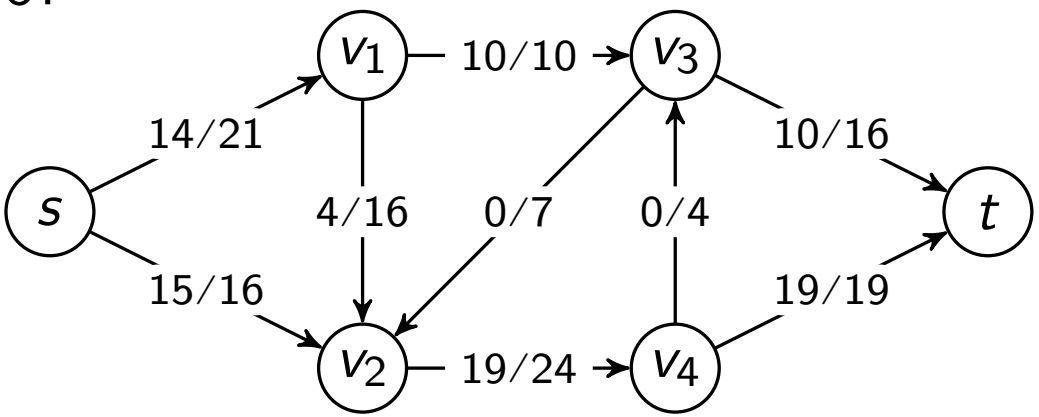
Oppgave 10: Hva er flyten over snittet $(\{s, v_2\}, \{v_1, v_3, v_4, t\})$ i flytnettverket etter å ha lagt til den flytforøkende fra oppgave 7 og 8?



$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

Maksimal flyt - Flytnett

Oppgave 10: Hva er flyten over snittet $(\{s, v_2\}, \{v_1, v_3, v_4, t\})$ i flytnettverket etter å ha lagt til den flytforøkende fra oppgave 7 og 8?

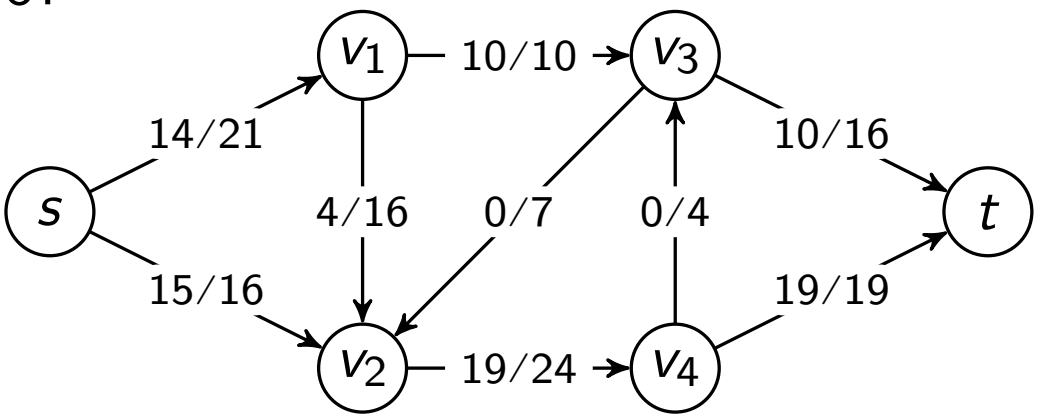


$$f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$$

$$f(\{s, v_2\}, \{v_1, v_3, v_4, t\}) = f(s, v_1) + f(v_2, v_4) - f(v_1, v_2) - f(v_3, v_2)$$

Maksimal flyt - Flytnett

Oppgave 10: Hva er flyten over snittet $(\{s, v_2\}, \{v_1, v_3, v_4, t\})$ i flytnettverket etter å ha lagt til den flytforøkende fra oppgave 7 og 8?



$$f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$$

$$\begin{aligned} f(\{s, v_2\}, \{v_1, v_3, v_4, t\}) &= f(s, v_1) + f(v_2, v_4) - f(v_1, v_2) - f(v_3, v_2) \\ &= 14 + 19 - 4 - 0 \\ &= 29 \end{aligned}$$

Edmonds-Karp

Oppgave 11: Hva er sammenhengen mellom Edmonds-Karp og Ford-Fulkerson?

Ford-Fulkerson(G, s, t)
1 initialize flow f to 0
2 **while** there exists an augmenting path p in the residual network G_f
3 augment flow f along p
4 **return** f

Edmonds-Karp

Oppgave 11: Hva er sammenhengen mellom Edmonds-Karp og Ford-Fulkerson?

Ford-Fulkerson(G, s, t)
1 initialize flow f to 0
2 **while** there exists an augmenting path p in the residual network G_f
3 augment flow f along p
4 **return** f

I Edmonds-Karp bruker vi BFS til å finne den forøkende stien p.

Oppgave 12: Hva stemmer om kjøretiden til Ford-Fulkerson?

- Ford-Fulkerson får dårlig kjøretid siden den finner den korteste forøkende stien, som kan medføre at vi kun øker flyten i flytnettet med én flytenhet for hver iterasjon av for-løkken.
- Lurespørsmål! Ford-Fulkerson er ikke en algoritme og kan derfor ikke ha en kjøretid.
- Fordi vi måler kjøretid basert på input-størrelse, og tallet $|f^*|$ kan lagres med $c \lg |f^*|$ bits, hvor c er konstant, blir kjøretiden eksponentiell som en funksjon av størrelsen.
- Ford-Fulkerson har polynomisk kjøretid, dersom vi definerer kjøretiden som en funksjon av verdien til $|f^*|$, og ikke input-størrelsen.

Oppgave 12: Hva stemmer om kjøretiden til Ford-Fulkerson?

- ~~Ford-Fulkerson får dårlig kjøretid siden den finner den korteste forøkende stien, som kan medføre at vi kun øker flyten i flytnettet med én flytenhet for hver iterasjon av for-løkken.~~
- Lurespørsmål! Ford-Fulkerson er ikke en algoritme og kan derfor ikke ha en kjøretid.
- Fordi vi måler kjøretid basert på input-størrelse, og tallet $|f^*|$ kan lagres med $c \lg |f^*|$ bits, hvor c er konstant, blir kjøretiden eksponentiell som en funksjon av størrelsen.
- Ford-Fulkerson har polynomisk kjøretid, dersom vi definerer kjøretiden som en funksjon av verdien til $|f^*|$, og ikke input-størrelsen.

Oppgave 12: Hva stemmer om kjøretiden til Ford-Fulkerson?

- ~~Ford-Fulkerson får dårlig kjøretid siden den finner den korteste forøkende stien, som kan medføre at vi kun øker flyten i flytnettet med én flytenhet for hver iterasjon av for-løkken.~~
- ~~Lurespørsmål! Ford-Fulkerson er ikke en algoritme og kan derfor ikke ha en kjøretid.~~
- Fordi vi måler kjøretid basert på input-størrelse, og tallet $|f^*|$ kan lagres med $c \lg |f^*|$ bits, hvor c er konstant, blir kjøretiden eksponentiell som en funksjon av størrelsen.
- Ford-Fulkerson har polynomisk kjøretid, dersom vi definerer kjøretiden som en funksjon av verdien til $|f^*|$, og ikke input-størrelsen.

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

Hvorfor? BFS finner den korteste stien fra s til alle andre noder i residualnettverket (basert på antall kanter).

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

Hvorfor? BFS finner den korteste stien fra s til alle andre noder i residualnettverket (basert på antall kanter).

Endringen i residualnettverket fra en slik forøkende sti vil ikke føre til kortere veier til noen noder.

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

Hvorfor? BFS finner den korteste stien fra s til alle andre noder i residualnettverket (basert på antall kanter).

Endringen i residualnettverket fra en slik forøkende sti vil ikke føre til kortere veier til noen noder.

For hver endring får vi minst en kritisk kant som forsvinner fra residualnettverket.

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

Hvorfor? BFS finner den korteste stien fra s til alle andre noder i residualnettverket (basert på antall kanter).

Endringen i residualnettverket fra en slik forøkende sti vil ikke føre til kortere veier til noen noder.

For hver endring får vi minst en kritisk kant som forsvinner fra residualnettverket.

En kant som har forsvunnet kan kun komme tilbake hvis det går flyt motsatt vei. Det kan kun skje hvis den korteste veien til noden den går til har økt med minst 2.

Oppgave 13: Hvorfor fører Edmonds-Karp valg av metode for å finne forøkende stier til at vi kan gi en garanti om at kjøretiden reduseres?

Metoden setter en begrensning på antall ganger vi kan finne flytforøkende stier.

En kant som har forsvunnet kan kun komme tilbake hvis det går flyt motsatt vei. Det kan kun skje hvis den korteste veien til noden den går til har økt med minst 2.

Hver kant kan kun forsvinne maksimalt $\frac{|\mathrm{V}|}{2}$ ganger, og vi har E kanter. Kan maksimalt ha EV iterasjoner.

Edmonds-Karp - Implementasjon

Oppgave 14: Implementer Edmonds-Karp

Bipartitt matching

Oppgave 15: Hva er en matching?

Bipartitt matching

Oppgave 15: Hva er en matching?

En delmengde av alle kanter, der hver node er tilknyttet maks en kant fra delmengden..

Bipartitt matching

Oppgave 15: Hva er en matching?

En delmengde av alle kanter, der hver node er tilknyttet maks en kant fra delmengden..

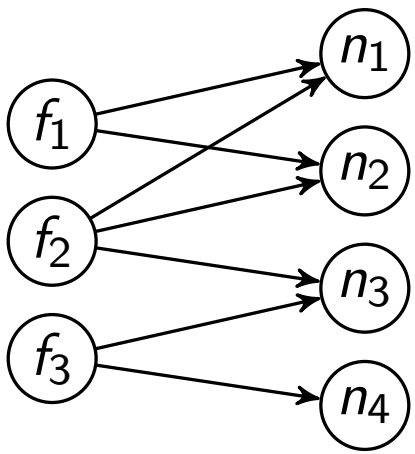
Oppgave 16: Et sett med filer har filnavn som må forkortes til 8 tegn. Filnavnene har hver et sett med meningsfylte forkortelser og det kan være overlapp mellom disse settene. Hvordan kan du løse dette problemet algoritmisk?

Bipartitt matching

Oppgave 15: Hva er en matching?

En delmengde av alle kanter, der hver node er tilknyttet maks en kant fra delmengden..

Oppgave 16: Et sett med filer har filnavn som må forkortes til 8 tegn. Filnavnene har hver et sett med meningsfylte forkortelser og det kan være overlapp mellom disse settene. Hvordan kan du løse dette problemet algoritmisk?



NPC

Oppgave 17: Hvilke utsagn stemmer?

- Dersom man kan løse ett problem i NPC i polynomisk tid så kan man løse alle problemer i NP i polynomisk tid.
- Alle problemene i NPC kan verifiseres i polynomisk tid.
- Om en algoritme løser et problem i NP, så kan den også løse alle NP-harde problemer.
- For at et problem skal være i NP må det kunne løses i polynomisk tid.

NPC

Oppgave 17: Hvilke utsagn stemmer?

- Dersom man kan løse ett problem i NPC i polynomisk tid så kan man løse alle problemer i NP i polynomisk tid.
- Alle problemene i NPC kan verifiseres i polynomisk tid.
- ~~Om en algoritme løser et problem i NP, så kan den også løse alle NP-harde problemer.~~
- For at et problem skal være i NP må det kunne løses i polynomisk tid.

Oppgave 17: Hvilke utsagn stemmer?

- Dersom man kan løse ett problem i NPC i polynomisk tid så kan man løse alle problemer i NP i polynomisk tid.
- Alle problemene i NPC kan verifiseres i polynomisk tid.
- ~~Om en algoritme løser et problem i NP, så kan den også løse alle NP-harde problemer.~~
- ~~For at et problem skal være i NP må det kunne løses i polynomisk tid.~~

Oppgave 18: Hvilke utsagn stemmer?

- Et optimeringsproblem er som regel enklere å løse enn et tilhørende bestemmelsesproblem.
- Klassen co-NP består av språkene L der komplementet av L kan verifiseres i polynomisk tid.
- Kodingen av en probleminstans har ingen ting å si for kompleksiteten av å løse problemet.
- En instans av et konkret problem er en binær string.
- En algoritme A sies å akseptere et språk L dersom $\mathrm{A}(x) = 1$ hvis $x \in \mathrm{L}$ og $\mathrm{A}(x) = 0$ hvis $x \notin \mathrm{L}$.

Oppgave 18: Hvilke utsagn stemmer?

- ~~Et optimeringsproblem er som regel enklere å løse enn et tilhørende bestemmelsesproblem.~~
- Klassen co-NP består av språkene L der komplementet av L kan verifiseres i polynomisk tid.
- Kodingen av en probleminstans har ingen ting å si for kompleksiteten av å løse problemet.
- En instans av et konkret problem er en binær string.
- En algoritme A sies å akseptere et språk L dersom $\mathrm{A}(x) = 1$ hvis $x \in \mathrm{L}$ og $\mathrm{A}(x) = 0$ hvis $x \notin \mathrm{L}$.

Oppgave 18: Hvilke utsagn stemmer?

- ~~Et optimeringsproblem er som regel enklere å løse enn et tilhørende bestemmelsesproblem.~~
- Klassen co-NP består av språkene L der komplementet av L kan verifiseres i polynomisk tid.
- ~~Kodingen av en probleminstans har ingen ting å si for kompleksiteten av å løse problemet.~~
- En instans av et konkret problem er en binær string.
- En algoritme A sies å akseptere et språk L dersom $\mathrm{A}(x) = 1$ hvis $x \in \mathrm{L}$ og $\mathrm{A}(x) = 0$ hvis $x \notin \mathrm{L}$.

Oppgave 18: Hvilke utsagn stemmer?

- ~~Et optimeringsproblem er som regel enklere å løse enn et tilhørende bestemmelsesproblem.~~
- Klassen co-NP består av språkene L der komplementet av L kan verifiseres i polynomisk tid.
- ~~Kodingen av en probleminstans har ingen ting å si for kompleksiteten av å løse problemet.~~
- En instans av et konkret problem er en binær string.
- ~~En algoritme A sies å akseptere et språk L dersom $\mathrm{A}(x) = 1$ hvis $x \in \mathrm{L}$ og $\mathrm{A}(x) = 0$ hvis $x \notin \mathrm{L}$.~~

Kompleksitetsklasser

Oppgave 19: Hva vet vi at stemmer om kjøretidsklassene?

- $\mathrm{P} \subseteq \mathrm{NP}$
- $\mathrm{P} = \mathrm{NP}$
- $\mathrm{P} \subseteq \text{co-NP}$
- $\text{NP-hard} \subseteq \mathrm{NP}$
- $\mathrm{NPC} \subseteq \text{NP-hard}$

Kompleksitetsklasser

Oppgave 19: Hva vet vi at stemmer om kjøretidsklassene?

- $\mathrm{P} \subseteq \mathrm{NP}$
- ~~$\mathrm{P} = \mathrm{NP}$~~
- $\mathrm{P} \subseteq \text{co-NP}$
- $\text{NP-hard} \subseteq \mathrm{NP}$
- $\mathrm{NPC} \subseteq \text{NP-hard}$

Kompleksitetsklasser

Oppgave 19: Hva vet vi at stemmer om kjøretidsklassene?

- $\mathrm{P} \subseteq \mathrm{NP}$
- ~~$\mathrm{P} = \mathrm{NP}$~~
- $\mathrm{P} \subseteq \text{co-NP}$
- ~~$\text{NP-hard} \subseteq \mathrm{NP}$~~
- $\mathrm{NPC} \subseteq \text{NP-hard}$

Det binære ryggsekkproblemet

Oppgave 20: Det binære ryggsekkproblemet er et NP-komplett problem, men vi kjenner en algoritme som kan løse det med kjøretid $\mathrm{O}(n\mathrm{W})$. Denne kjøretiden kalles pseudopolynomisk. Hva betyr dette?

Det binære ryggsekkproblemet

Oppgave 20: Det binære ryggsekkproblemet er et NP-komplett problem, men vi kjenner en algoritme som kan løse det med kjøretid $\mathrm{O}(n\mathrm{W})$. Denne kjøretiden kalles pseudopolynomisk. Hva betyr dette?

Vi representerer tall som et sett med bits. Algoritmen er avhengig av antall bits w som trengs for å representere W.

Det binære ryggsekkproblemet

Oppgave 20: Det binære ryggsekkproblemet er et NP-komplett problem, men vi kjenner en algoritme som kan løse det med kjøretid $\mathrm{O}(n\mathrm{W})$. Denne kjøretiden kalles pseudopolynomisk. Hva betyr dette?

Vi representerer tall som et sett med bits. Algoritmen er avhengig av antall bits w som trengs for å representere W.

Kjøretiden blir da $\mathrm{O}(n2^w)$.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$.
- Finne ut om en graf har en sti mellom to noder med lengde $> k$.
- Finne ut om en graf har en hamiltonsykel.
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar.
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar.
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$.
- Finne ut om en graf har en hamiltonsykel.
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar.
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar.
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel.
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar.
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar.
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel. **NP, NP-hardt, NPC**
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar.
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar.
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel. **NP, NP-hardt, NPC**
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar. **P, NP**
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar.
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel. **NP, NP-hardt, NPC**
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar. **P, NP**
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar. **NP, NP-hardt, NPC**
- Finne den største mulige klikken i en graf.
- Finne ut om en digital krets kan gi 1 som output.
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel. **NP, NP-hardt, NPC**
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar. **P, NP**
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar. **NP, NP-hardt, NPC**
- Finne den største mulige klikken i en graf. **NP-hardt** *[Endret etter videoen ble laget]*
- Finne ut om en digital krets kan gi 1 som output. **NP, NP-hardt, NPC**
- Finne ut om et program kommer til å stoppe eller kjøre for alltid.

Kjente problemer - Kompleksitetsklasse

Oppgaver 21, 22, 23 og 24: Hvilke av disse problemene er NP-harde og/eller hører til P, NP og/eller NPC?

- Sortere en liste med elementer.
- Finne ut om en graf har en sti mellom to noder med lengde $\leqslant k$. **P, NP**
- Finne ut om en graf har en sti mellom to noder med lengde $> k$. **NP, NP-hardt, NPC**
- Finne ut om en graf har en hamiltonsykel. **NP, NP-hardt, NPC**
- Finne ut om en boolsk 2-CNF-formel er oppfyllbar. **P, NP**
- Finne ut om en boolsk 3-CNF-formel er oppfyllbar. **NP, NP-hardt, NPC**
- Finne den største mulige klikken i en graf. **NP-hardt** *[Endret etter videoen ble laget]*
- Finne ut om en digital krets kan gi 1 som output. **NP, NP-hardt, NPC**
- Finne ut om et program kommer til å stoppe eller kjøre for alltid. **NP-hardt**

Reduksjon

Oppgave 25: Gitt et problem A som er bevist å være i NP og et problem B som er bevist NP-hardt, hva stemmer?

- Hvis man finner en polynomisk reduksjon fra B til A har man vist at $P = NP$
- B er i NPC hvis man kan verfisere B i polynomisk tid.
- Hvis man finner en polynomisk reduksjon fra B til A har man vist at A er i NPC.
- Hvis man finner en polynomisk reduksjon fra A til B har man vist at A er i NPC.
- Hvis man finner en polynomisk reduksjon fra A til B har man vist at $P = NP$.
- A er i NPC hvis man kan verifisere A i polynomisk tid.

Reduksjon

Oppgave 25: Gitt et problem A som er bevist å være i NP og et problem B som er bevist NP-hardt, hva stemmer?

- ~~Hvis man finner en polynomisk reduksjon fra B til A har man vist at $P = NP$~~
- B er i NPC hvis man kan verfisere B i polynomisk tid.
- Hvis man finner en polynomisk reduksjon fra B til A har man vist at A er i NPC.
- Hvis man finner en polynomisk reduksjon fra A til B har man vist at A er i NPC.
- Hvis man finner en polynomisk reduksjon fra A til B har man vist at $P = NP$.
- A er i NPC hvis man kan verifisere A i polynomisk tid.

Reduksjon

Oppgave 25: Gitt et problem A som er bevist å være i NP og et problem B som er bevist NP-hardt, hva stemmer?

- ~~Hvis man finner en polynomisk reduksjon fra B til A har man vist at $P = NP$~~
- B er i NPC hvis man kan verfisere B i polynomisk tid.
- Hvis man finner en polynomisk reduksjon fra B til A har man vist at A er i NPC.
- ~~Hvis man finner en polynomisk reduksjon fra A til B har man vist at A er i NPC.~~
- Hvis man finner en polynomisk reduksjon fra A til B har man vist at $P = NP$.
- A er i NPC hvis man kan verifisere A i polynomisk tid.

Reduksjon

Oppgave 25: Gitt et problem A som er bevist å være i NP og et problem B som er bevist NP-hardt, hva stemmer?

- ~~Hvis man finner en polynomisk reduksjon fra B til A har man vist at $P = NP$~~
- B er i NPC hvis man kan verfisere B i polynomisk tid.
- Hvis man finner en polynomisk reduksjon fra B til A har man vist at A er i NPC.
- ~~Hvis man finner en polynomisk reduksjon fra A til B har man vist at A er i NPC.~~
- ~~Hvis man finner en polynomisk reduksjon fra A til B har man vist at $P = NP$.~~
- A er i NPC hvis man kan verifisere A i polynomisk tid.

Reduksjon

Oppgave 25: Gitt et problem A som er bevist å være i NP og et problem B som er bevist NP-hardt, hva stemmer?

- ~~Hvis man finner en polynomisk reduksjon fra B til A har man vist at $P = NP$~~
- B er i NPC hvis man kan verfisere B i polynomisk tid.
- Hvis man finner en polynomisk reduksjon fra B til A har man vist at A er i NPC.
- ~~Hvis man finner en polynomisk reduksjon fra A til B har man vist at A er i NPC.~~
- ~~Hvis man finner en polynomisk reduksjon fra A til B har man vist at $P = NP$.~~
- ~~A er i NPC hvis man kan verifisere A i polynomisk tid.~~

NPC-bevis

Oppgave 26: La A være problemet om å bestemme om en urettet graf har en stabil mengde bestående av k noder. Bevis at A er NP-komplett.

NPC-bevis

Oppgave 26: La A være problemet om å bestemme om en urettet graf har en stabil mengde bestående av k noder. Bevis at A er NP-komplett.

A er i NP fordi vi kan verifisere en løsning med $\mathrm{O}(\mathrm{EV})$ tidskompleksitet.

NPC-bevis

Oppgave 26: La A være problemet om å bestemme om en urettet graf har en stabil mengde bestående av k noder. Bevis at A er NP-komplett.

A er i NP fordi vi kan verifisere en løsning med $\mathrm{O}(\mathrm{EV})$ tidskompleksitet.

Vi kan redusere fra CLIQUE-problemet ved å finne komplementærgrafen. Det vil si A er NP-hard.

Verifisere handelsreiseproblemet - Programmering

Oppgave 27: Skriv en funksjon som kan verifisere handelsreiseproblemet.

Verifisere handelsreiseproblemet - Programmering

Oppgave 27: Skriv en funksjon som kan verifisere handelsreiseproblemet.

1. Sjekk om stien starter og slutter i samme node.
2. Sjekk om stien går igjennom de resterende nodene nøyaktig en gang.
3. Sjekk om kostanden til stien er mindre eller lik den oppgitte maksimale kostnaden k.