

i **Informasjon om prøveeksamen**

Velkommen til prøveeksamen i IN2010/INF2220 - Algoritmer og datastrukturer.

Tid: 21.-27. november 2019

**Tillatte hjelpemidler: ingen.**

Det anbefales å lese raskt gjennom eksamen før du setter i gang, for å få ett overblikk (og for å notere eventuelle spørsmål til faglærer).

Oppgavesettet består av tre seksjoner som er vektet omtrent slik:

- **33%** for seksjon 1 (korte svar, automatisk rettet)
- **27%** for seksjon 2 (diverse programmering)
- **40%** for seksjon 3 (agentnett)

For hver oppgave er maksimal poengsum oppgitt. Dette indikerer vanskelighetsgraden/viktigheten av oppgaven. Ikke bruk alt for mye tid på en oppgave med 1-2 poeng.

I oppgave **3e** skal du svare med digital håndtegning. Du skal bruke skisseark som du får utdelt. Det er **ikke** anledning til å bruke bruke digital håndtegning på andre oppgaver enn **3e**. Det blir **ikke** gitt ekstratid for å fylle ut informasjonsboksene på skisseark (engangskoder, kand.nr. o.l.).

For grafoppgavene kan du anta at grafen ikke inneholder

- enkle løkker (en kant fra en node til seg selv)
- parallelle kanter (mer enn en kant mellom et par av noder)

For oppgaver om trær skal du anta at et tre med bare én node har høyde 0.

For oppgaver om  $P$  og  $NP$  skal du anta at  $P \neq NP$ .

For oppgaver som ber om kode (**2a-2d**), kan du svare med pseudokode/naturlig språk, men jo næremer en fullt detaljert algoritme svaret ditt er, jo bedre. I **3g** er valg av beskrivelse helt opp til deg, selv om du er gitt en kode-editor som svarfelt.

1(a) **Separasjonsnoder i spenltrær**

Den urettede sammenhengende grafen  $G$  har 12 noder og 20 kanter. La  $S$  være et spenntre for  $G$ . Avhengig av hvordan  $G$  ser ut, hva er det minste antall separasjonsnoder  $S$  kan inneholde?

Svar her: .

Maks poeng: 3

1(b) **Trikkestopp**

Bob skal reise med trikken. Bob bor rett utenfor trikkestoppet Alpha.  
Trikkesystemet i byen der Bob bor, fungerer slik at man betaler 5 kroner for hvert stopp man reiser med trikken.  
Bob ønsker å bergene hvor mye det vil koste ham å reise med trikken fra Alpha til alle andre trikkestopp i byen.

Hvilken av følgende algoritmer er den **raskeste** algoritmen Bob kan bruke for å finne ut hva det vil koste å reise til hvert av de andre stoppene fra Alpha, gitt en grafrepresentasjon av trikkesystemet i byen.

Velg ett alternativ

- ☐ Dijkstra
- ☐ Topologisk sortering
- ☐ Bredde-først-søk
- ☐ Bellman-Ford

Maks poeng: 4

1(c) **P og NP**

Hvis vi antar at  $P \neq NP$ , så kan ingen  $NP$ -komplette problemer løses i polynomiell tid.

Velg ett alternativ

- ☐ Usant
- ☐ Sant

Maks poeng: 2

1(d) **Splitt og hersk**

Hvilke sorteringsalgoritmer benytter en splitt-og-hersk strategi?

Velg ett eller flere alternativer

- ☐ radix
- ☐ flette
- ☐ quick
- ☐ utplukk
- ☐ heap
- ☐ innstikk

Maks poeng: 2

1(e) **Stabile sorteringsalgoritmer**

Hvilke av disse sorteringsalgoritmene er stabile?

**Velg ett eller flere alternativer**

- ☐ utplukkssortering
- ☐ heapsortering
- ☐ innstikkssortering
- ☐ flettesortering
- ☐ kvikksortering

Maks poeng: 3

1(f) **Heap i array**

Vi har en heap i en array. Rota ligger på indeks 0. Siste node i heapen ligger på indeks 40.

Hva er høyden til heapen?

Svar her: .

Hvilken indeks ligger forelderen til siste node på?

Svar her: .

Er noden på indeks 40 venstrebarnd eller høyrebarnd?

Svar her (0 for venstrebarnd, 1 for høyrebarnd): .

Maks poeng: 3

1(g) **Binære søketrær I**

Vi setter inn elementer med følgende verdier inn i et binært søketre i denne rekkefølgen:

23, 45, 12, 19, 32, 67, 35

Hvilken verdi har rota i treet?

Svar her:  .

Hvor mange noder har subtreet som har noden med verdi **12** som rot?

Svar her:  .

Hva er høyden til treet?

Svar her:  .

Maks poeng: 3

1(h) **Binære søketrær II**

Vi setter inn elementer med følgende verdier inn i et binært søketre:

67, 45, 35, 32, 23, 19, 12

Hva er høyden til treet?

Svar her:  .

Maks poeng: 1

1(i) **Binære søketrær III**

Vi setter **17** elementer inn i et binært søketre.

Hva er den minimale høyden til treet?

Svar her:  .

Og hva er den maksimale høyden?

Svar her:  .

Maks poeng: 2

1(j) **Fjerne node i et binært søketre**

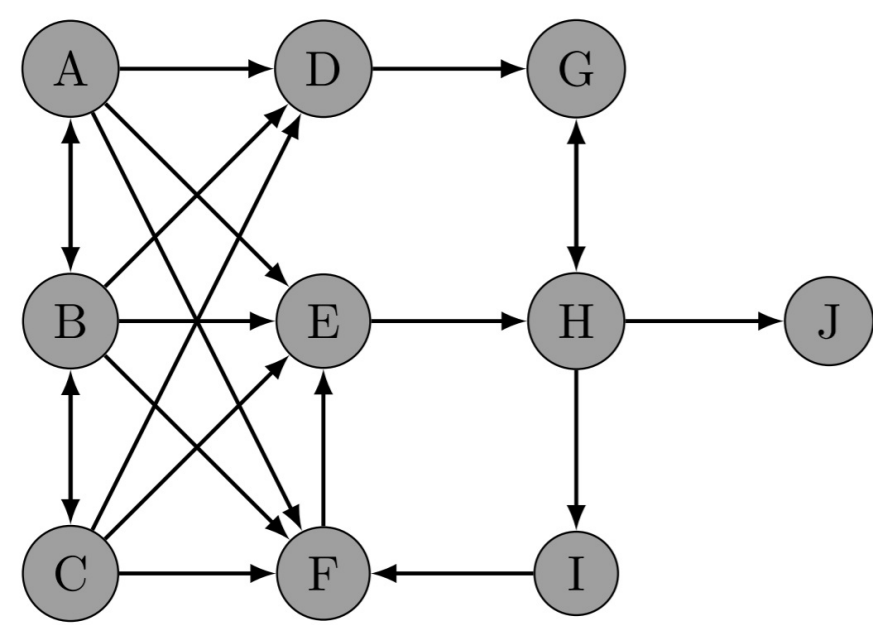
La  $B$  være et binært søketre med mindre verdier til venstre og større eller like verdier til høyre.  
Vi skal fjerne en node  $v$  i  $B$ . Noden  $v$  har to ikke-tomme subtrær.  
Hvilken node i  $B$  skal erstatte  $v$ ?

Velg ett alternativ

- ☐ noden i  $v$  sitt høyre subtre som har størst verdi
- ☐ noden i  $v$  sitt venstre subtre som har minst verdi
- ☐ rota i  $B$
- ☐ noden i  $v$  sitt høyre subtre som har minst verdi
- ☐ noden med størst verdi i  $B$
- ☐ noden med minst verdi i  $B$
- ☐ ingen

Maks poeng: 2

1(k) **Sterkt sammenhengende komponenter I**



Hva er de sterkt sammenhengende komponentene til grafen på bildet?  
Svar med en kommaseparert liste, for eksempel: AB, CD, EF, GH, IJ.

Svar her: .

Maks poeng: 2

1(l) **Sterkt sammenhengende komponenter II**

La  $G$  være en sammenhengende rettet graf der alle noder har inngrad  $> 0$  og utgrad  $> 0$ .

Vi vet at  $G$  kun har én sterkt sammenhengende komponent.

Velg ett alternativ

- ☐ Sant
- ☐ Usant

Maks poeng: 1

1(m) **Krav til rød-svarte trær**

Vi har følgende krav til et rød-svart tre

- Treet må være et .
- Rota må være .
- Ingen  noder skal ha  barn.
- Alle stier fra  skal inneholde like mange  noder.

Maks poeng: 5

**Java- eller pseudokode:**

7/16

```
class Beholder {
    // ... intern struktur og private klasser og metoder ...
    public Beholder ( int [ ] A ); // oppretter en beholder med elementene fra A
    public void push(int i); // legger elementet med verdi i inn i beholderen
    public int pop(); // tar ut et element av beholderen
    public int size(); // antall elementer i beholderen
}
```

```
public Beholder concatenate( Beholder a, Beholder b )
```

Programmer kvikksortering (quick sort). Du kan fritt bruke metodene beskrevet ovenfor ved å legge arrayen inn i en beholder og bruke metodene (uten å måtte programmere dem), eller operere direkte på arrayen. Hvis algoritmen din forutsetter noe mer om hvordan beholderen virker, skriv det inn som kommentarer i programmet.

**Javakode:**

8/16



2(c) **Huffmankoding: assignCode**

Gitt klassen som beskriver en *frekvensnode*:

```
class Node {
    Node left;
    Node right;
    char c; // tegnet
    int f; // frekvensen
    String code; // huffmankoden til tegnet.
            // Det er denne som skal få verdi med assignCode
}
```

Vi har kjørt huffmanalgoritmen på en tekststreng bestående av mer enn ett unikt tegn og bygget et huffmantre bygget opp av noder av klassen gitt ovenfor. Treet pekes på av en variabel Node rot.

*Du kan teste om noden n har et tegn assosiert med seg med if ( n.c == 0 ), men oppgaven lar seg løse uten å teste på dette attributtet. Skriv en metode i Java*

```
void assignCode(Node n, String s)
```

som gjør at et kall på assignCode(rot, "") setter korrekt huffmankode i attributtet *code* i alle noder som har et tegn assosiert med seg.

**Javakode:**

1

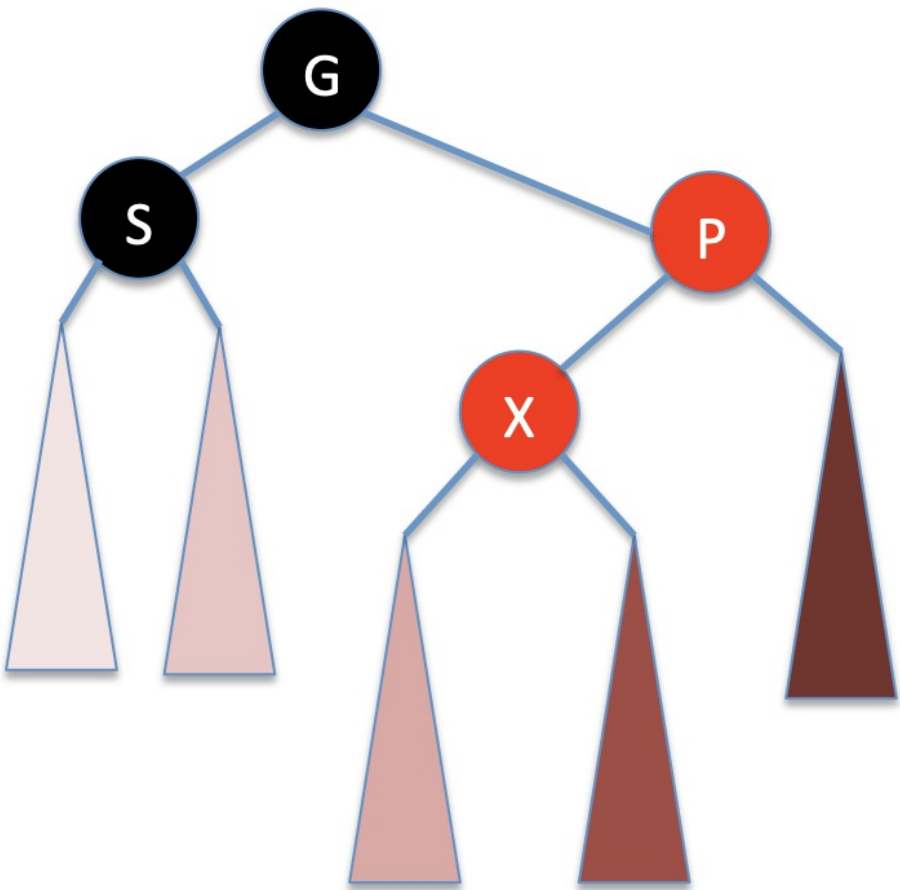
Maks poeng: 8

2(d) **Rotasjon i rød-svart tre**

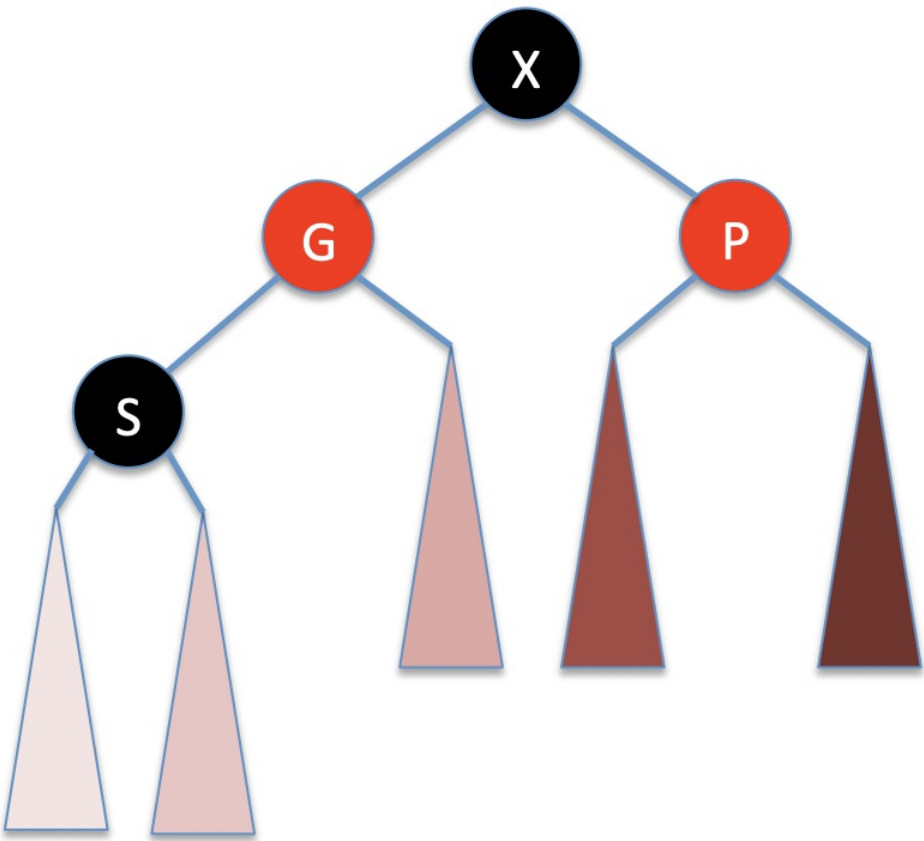
Gitt klassen som beskriver en *node i et rød-svart tre*:

```
class Node {
    Node parent;
    Node left;
    Node right;
```

```
Prøveeksamen IN2010 høsten 19
Char c; // farge = 'B' eller 'R'
}
```



Lag en metode som tar inn X som parameter og som gjør denne rotasjonen og nødvendige omfarginger, slik at resultatet etter kallet blir:



1	
---	--

Maks poeng: 7

i **Introduksjon**

Denne seksjonen er delt inn i åtte oppgaver (a-h) som delvis bygger på hverandre.  
Hvis du står fast på en oppgave er det mulig å gå videre til neste, men det er sterkt anbefalt å gjøre dem i den rekkefølgen de er gitt.  
Under følger viktig informasjon om oppgaven som gjelder gjennom hele seksjonen.

Du er ansatt som etterretningssjef i den nasjonale etterretningstjenesten.  
Din oppgave er blant annet å administrere agenter/spioner som er plassert rundt om i verden.  
Et **agentnettverk/agentnett** er en samling av agenter sammen med trygge kommunikasjonskanaler mellom par av agenter som muliggjør deling av viktig informasjon.

Vi representerer et agentnettverk som en urettet, vektet graf  $G = \langle V, E \rangle$ , der nodene representerer agentene. Det går en kant mellom to noder hvis agentene de representerer har en felles kommunikasjonskanal. Hver kant merkes med et heltall som angir relativ risiko for lekkasje ved å bruke kanalen. Kanalen med lavest risiko får verdi 1, den med nest lavest risiko får verdi 2 osv. Kanalen med høyest risiko har verdi  $|E|$  (antall kanter i grafen).

For enkelhets skyld skal du anta at det maks finnes én kommunikasjonskanal mellom to agenter og at ingen agenter har en kanal til seg selv.

3(a) **Minimumskrav**

Som etterretningssjef ønsker du at agentnettet skal oppfylle følgende **minimumskrav**:

- Alle agenter skal kunne dele informasjon med hverandre (muligens via andre agenter i nettverket).

For eksempel kan Oscar dele informasjon med Vesper ved å bruke en kommunikasjonskanal til James som videre bruker en kommunikasjonskanal til Vesper.

Gitt et agentnettverk representert ved  $G = \langle V, E \rangle$ , hvilken algoritme kjent fra IN2010 ønsker du å benytte for å sjekke om nettverket oppfyller minimumskravet?

1. Gi en kort begrunnelse (1-3 setninger) for hvorfor algoritmen gir deg det du er ute etter.
2. Oppgi kjøretiden til den valgte algoritmen i O-notasjon. Her trenger du ikke gi en begrunnelse.

**Skriv ditt svar her:**

Maks poeng: 4

3(b) **Mellomagenter**

Hvis et agentnett ikke oppfyller minimumskravet er en mulig løsning å innføre **mellomagenter** (med tilhørende kommunikasjonskanaler), som skal forsikre at alle kan dele informasjon med hverandre.

Forklar hvorfor det (i teorien) holder å innføre kun én mellomagent uansett hvordan agentnettet representert ved  $G$  ser ut.

**Skriv ditt svar her:**

Maks poeng: 2

3(c) **Nye kommunikasjonskanaler**

I praksis er det ikke alltid like lett å innføre nye mellomagenter for å oppfylle minimumskravet.

Du ønsker heller å håndtere minimumskravet ved å åpne nye kommunikasjonskanaler mellom par av agenter som tidligere ikke kunne dele informasjon med hverandre. Siden alle kanaler innebærer en viss risiko, ønsker du å legge til så få nye kommunikasjonskanaler som mulig.

Hvordan vil du regne ut det minimale antall nye kommunikasjonskanaler som må åpnes?  
Her kan du om ønskelig utvide algoritmen fra deloppgave a. Forklar med ord hva du ønsker å gjøre.

**Skriv ditt svar her:**

Maks poeng: 5

3(d) **Ordrekanaler I**

Du har nå sørget for at agentnettverket, representert ved  $G = \langle V, E \rangle$ , oppfyller minimumskravet. Det neste du ønsker er en måte å kunne gi ordre til alle agenter på, med lavest mulig risiko for lekkasje.  
Du har tenkt til å få til dette ved å oppgradere noen av kommunikasjonskanalene til **ordrekanaler**, altså kommunikasjonskanaler der ordre kan sendes.

En ordre skal nå alle agenter på følgende måte:

1. Du gir ordenen til Oscar, din mest betrodde agent.
2. Oscar videreformidler ordenen via alle sine ordrekanaler
3. Når en ny agent får høre ordenen, sender agenten ordenen videre via alle sine *andre* ordrekanaler (agenten sender altså ikke ordenen direkte tilbake via kanalen han fikk den fra, siden agenten på andre siden nødvendigvis allerede må ha mottatt ordenen).

**Merk:** Punkt 3 gjelder for alle agenter som ikke er Oscar.

Du ønsker, som nevnt over, at alle agentene skal motta ordenen. Hele prosessen skal gjøres med så lav total risiko som mulig.  
Utfordringen er nå å finne ut hvilke kommunikasjonskanaler som skal oppgraderes til ordrekanaler.

I IN2010 har vi lært tre algoritmer som potensielt kan løse dette problemet.

- Hvilke algoritmer er det snakk om?
- Vil alle tre fungere?
- Hvis flere av algoritmene fungerer, vil de oppgradere de samme ordrekanalene?

Gi en kort begrunnelse for hvert punkt.  
**Skriv ditt svar her:**

Maks poeng: 5

3(e)    **Ordrekanaler II**

Gitt følgende agentnett, bruk en av algoritmene fra kurset til å finne hvilke kanaler som skal oppgraderes til orderkanaler. Husk at du ønsker så lav total risiko som mulig.

Du skal tegne en graf som representerer agentnettet der kommunikasjonskanalene som er ordrekanaler er tydelig markert. Det er altså ikke nødvendig å gi alle steg i algoritmen, kun resultatet. Skriv også på tegningen hvilken algoritme du benyttet.

- Agenter:**
- Oscar (O)
  - James (J)
  - Kim (K)
  - Peter (P)
  - Cyclone (C)
  - Severus (S)
  - Ethan (E)
  - Vesper (V)
- Kommunikasjonskanaler:**
- Risiko 1: J-V
  - Risiko 2: P-S
  - Risiko 3: J-K
  - Risiko 4: K-V
  - Risiko 5: O-C
  - Risiko 6: K-E
  - Risiko 7: S-V
  - Risiko 8: S-E
  - Risiko 9: J-C
  - Risiko 10: O-J

Maks poeng: 7

3(f)    **Kompromitterte agenter**

Nå som du vet at agentnettverket oppfyller minimumskravet, ønsker du noe litt sterkere.

En uheldig men sannsynlig hendelse er at en agent blir kompromittert (satt ut av spill/avslørt). En kompromittert agent må fjernes fra nettverket sammen med alle kommunikasjonskanaler agenten hadde. Du ønsker nå et nettverk som oppfyller minimumskravet selv hvis en vilkårlig agent blir kompromittert. Et slikt nettverk kaller vi **kompromitteringstolerant**.

Gitt et agentnettverk representert ved  $G = \langle V, E \rangle$  som oppfyller minimumskravet, hvilken algoritme fra kurset ønsker du å benytte for å sjekke om nettverket er kompromitteringstolerant?

- Forklar hvorfor algoritmen gir deg det du er ute etter
- Hva er kjøretiden til algoritmen i O-notasjon? Her trenger du ikke oppgi begrunnelse

**Merk:** Svarene dine skal gjelde for et generelt agentnettverk ikke kun eksemplet fra deloppgave e.

**Skriv ditt svar her:**

Maks poeng: 4

3(g) **Ordrekritiske agenter**

Du har gjort nødvendige endringer i nettverket slik at det nå er kompromitteringstolerant.  
Du har også regnet ut ordrekanalene på nytt, siden nettverket har blitt endret.  
La grafen  $G = \langle V, E \rangle$  representere det nye nettverket.

En agent er **ordrekritisk** hvis det er slik at om agenten blir kompromittert, vil du ikke lenger kunne sende ordre til alle de resterende agentene ved å bruke ordrekanalene.  
Siden Oscar er så viktig i ordreformidlingen er han alltid å regne som ordrekritisk.

Skriv en algoritme, som gitt et kompromitteringstolerant agentnettverk  $G = \langle V, E \rangle$ , der  $Oscar \in V$ , finner alle ordrekritiske agenter (inkludert Oscar). Du kan anta at ordrekanalene på forhånd har blitt regnet ut (i henhold til kravene) og at alle kanter i  $E$  inneholder en boolsk variabel som indikerer om kanalen er en ordrekanal eller ikke.

Du kan bruke naturlig språk og/eller pseudokode for å beskrive algoritmen din.  
I tillegg er det mulig å gjenbruke algoritmer vi har sett på i IN2010. Hvis du for eksempel ønsker å traversere grafen med bredde-først søk, trenger du ikke å forklare hvordan bredde-først søk fungerer. Hvis du ønsker å benytte en modifisert versjon av en algoritme fra kurset, må modifikasjonene komme tydelig frem.

I IN2010 har vi lagt vekt på effektive algoritmer. I denne oppgaven vil derfor en rask algoritme gi større uttelling enn en mindre rask algoritme.  
I neste oppgave blir du bedt om å analysere kjøretiden til algoritmen din.

**Merk:** Svaret ditt skal gjelde for et generelt agentnettverk, ikke kun eksemplet fra deloppgave e.

Skriv ditt svar her:

1	
---	--

Maks poeng: 8

3(h) **Analyse**

Gi en analyse av kjøretiden til algoritmen din fra deloppgave g ved hjelp av O-notasjon.  
Hvis du brukte algoritmer kjent fra kurset trenger du ikke forklare hvorfor de har den kjøretiden de har.

Her vil du få uttelling om analysen din samsvarer med algoritmen du ga, uavhengig av hvor rask algoritmen er.

**Merk:** Svaret ditt skal gjelde for et generelt agentnettverk, ikke kun eksemplet fra deloppgave e.  
**Skriv ditt svar her...**

Maks poeng: 5