IN2010 uke 8

Jakob Hansen

14. oktober 2020

Hva vi kan snakke om idag

- Repetisjon
- Noen nye begreper for grafer
- Sære algoritmer
  - Finne separasjonsnoder i en 2-sammenhengende graf (Bikonnektivitet)
  - Finne sterkt sammenhengende komponenter
- Se på en større eksamensoppgave

Litt nye begreper

- Sammenhengende
- k-sammenhengende graf og bikonnektivitet
- Separasjonsnoder
- Sterkt sammenhengende komponenter

Finne ut om en graf er 2-sammenhengende

- Er grafen fortsatt sammenhengende om en node fjernes?
- Naiv algoritme: Fjern en node, sjekk om grafen fortsatt er sammenhengende -> $O(|V| * (|V| + |E|)$
- Bedre algoritme "Hopcroft-Tarjan"
  - Gjør DFS, sjekk om det er mulig å komme "tilbake" til grafen en annen vei enn der DFS kom fra
  - Bruker indekser og "low-verdier".
  - $O(|V| + |E|)$

Finne sterkt sammenhengende komponenter

- A -> B, B -> A
- Hvordan finne ut om det finnes en sti fra en node A til alle andre?
- DFS
- Hvordan finne ut om det finnes en sti mellom alle andre noder til A?
- Reverser grafen og kjør DFS fra A igjen!
- Alle noder som kan nås i begge retninger, er sterkt sammenhengende

Tarjan's lineær tid algoritme for SCC

- Kanskje et tilfelle av "ingen skjønner hvorfor dette fungerer"
- Fremgangsmåte:
  - Kjør DFS på alle noder, legg alle noder på en stack
  - Reverser grafen
  - Kjør DFS igjen, med stack rekkefølgen.
  - Alle noder som kan nås hver gang man kaller på DFS den andre gangen, er i samme SCC.

Eksamensoppgave

I denne oppgaven skal du hjelpe til å designe verdener i et dataspill.

En (spill)verden består av forskjellige rom. I hvert rom finnes det et panel som viser spilleren hvilke andre rom man kan gå til fra det aktuelle rommet.

Eksempelverden

Rom	A	B	C	D	E	F	G	H
Panel	B,C,F	A,H	A,D,E	G	F,G,H	E,H	H	

Fra et rom velger spilleren et nytt rom fra panelet. Når et rom er valgt, blir spilleren gitt en oppgave som må løses for å komme seg til rommet han valgte. Etter oppgaven er løst blir spilleren transportert til det valgte rommet.

Målet i hver verden er å komme frem til et **skattkammer** på kortest mulig tid.
Et rom k er et **skattkammer** hvis og bare hvis det oppfyller følgende krav:

- Det er ikke mulig å nå noen andre rom fra k.
- Det er mulig å komme seg til k fra alle andre rom i verdenen.

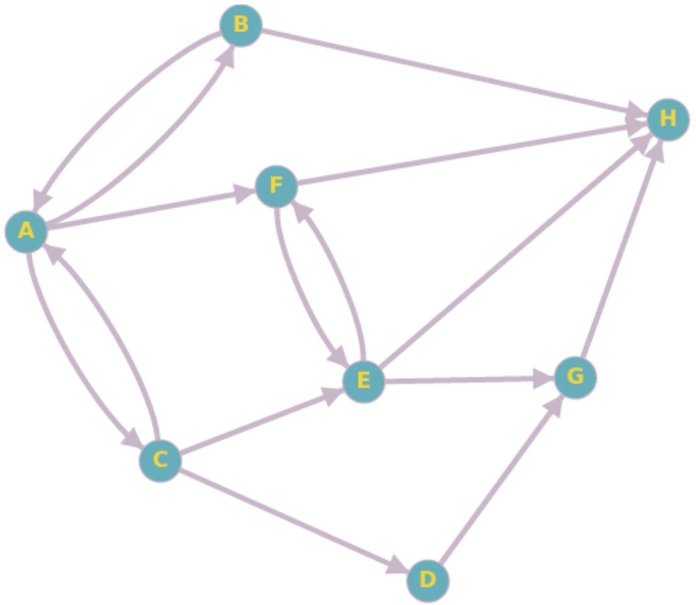
I eksempelverdenen over er H et skattkammer.

Hver verden begynner med at spilleren blir plassert i et vilkårlig **startrom**.
Et rom s er et **startrom** hvis og bare hvis det oppfyller følgende krav:

- Det er mulig å komme seg fra s til alle andre rom i verdenen.

I eksempelverdenen over er det tre startrom, nemlig A,B og C.

Eksempelgraf



2a

2(a) Representasjon

Vi skal modellere en verden som en rettet graf der nodene representerer rommene i verdenen.

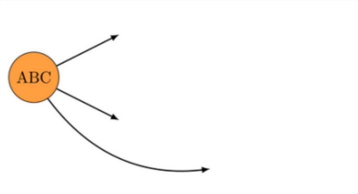
1. Hva utgjør kantene i grafen?
2. Hva bestemmer utgraden til en node?
3. Hva er utgraden til et skattkammer?
4. I eksempelverdenen, hva er utgraden til rom D?

Merk: Svarene dine skal gjelde for en generell verden, ikke bare eksempelverdenen. Dette gjelder ikke for punkt (4).

2b

2(b) **Komponentgraf**

Du blir gitt en verden og representerer den som en rettet graf $G = \langle V, E \rangle$ som i oppgave a.
Vi kan regne ut de sterkt sammenhengende komponentene til G og lage **komponentgrafen** C til G.
Komponentgrafen C er en graf der hver sterkt sammenhengende komponent i G utgjør nodene i C. Det går en rettet kant i C fra en komponent til en annen, hvis det gikk en kant, i G, fra en node i den første komponenten til en node i den andre komponenten.



Du får her en påbegynt tegning av komponentgrafen C til eksempelverdenen.
Legg merke til at {A, B, C} er en node i C, fordi {A, B, C} er en sterkt sammenhengende komponent i G, og at den noden har utgrad 3.
Fullfør tegningen av komponentgrafen til eksempelverdenen.

I denne oppgaven skal du svare med digital håndtegning. Bruk eget skisseark (utdelt). Se instruksjon for utfylling av skisseark i lenken under oppgavelinjen.

Hint: Det kan være lurt (med tanke på denne og de neste oppgavene) å også tegne grafen til eksempelverdenen for din egen del. En slik tegning skal *ikke* leveres inn.

Maks poeng: 3

2c

2(c) **Skattkammer I**

Du blir gitt en verden og representerer den som en rettet graf $G = \langle V, E \rangle$ som i oppgave a.
La noden $k \in V$ være et skattkammer.
La C_k være den sterkt sammenhengende komponenten til k.
For hvert av spørsmålene under, gi en kort begrunnelse (1-3 setninger).

1. Hvor mange noder inneholder C_k?
2. Hva er utgraden til C_k?

Merk: Svarene dine skal gjelde for en generell verden, ikke bare eksempelverdenen.

2d

2(d) Skattkammer II

Forklar kort hvorfor en verden ikke kan inneholde mer enn ett skattkammer.

Merk: Svaret ditt skal gjelde for en generell verden, ikke bare eksempelverdenen.

2e

2(e) Startrom

Du blir gitt en verden og representerer den som en rettet graf $G = \langle V, E \rangle$ som i oppgave a.
La noden $s \in V$ være et startrom.
La C_s være den sterkt sammenhengende komponenten til s. Vi skal nå vise at startrommene i verdenen er nøyaktig nodene i C_s.

1. Forklar hvorfor det er slik at hvis $t \in V$ også er i C_s, så må t være et startrom.
2. Vis at alle startrom i verdenen er i C_s.

Merk: Svarene dine skal gjelde for en generell verden, ikke bare eksempelverdenen.

2f

2(f)

Algoritmedesign

En spilldesigner har laget et forslag til en verden og representert den som en rettet graf $G = \langle V, E \rangle$. Du blir bedt om å sjekke om verdenen representert ved G har følgende egenskaper:

- verdenen har nøyaktig ett skattkammer
- verdenen har nøyaktig tre startrom.

Skriv en algoritme som sjekker om en foreslått verden har de to egenskapene over.

For enkelhetsskyld kan du anta at du får inn grafen G som input.

Du kan også anta at du for hver node får en liste over utgående- og innkommende kanter for noden.

Her kan du, om ønskelig, gjenbruke resultater fra de tidligere deloppgavene (selv de du ikke har besvart).

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 dybde-først søk, trenger du ikke å forklare hvordan dybde-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.

2g

2(g) **Analyse**

Gi en analyse av kjøretiden til algoritmen din fra deloppgave f 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 en generell verden, ikke bare eksempelverdenen.

2h

2(h) Minimal gjennomføringstid

I denne deloppgaven får du også oppgitt den minimale tiden som trengs for å løse de forskjellige oppgavene.

Panelene består nå av par av rom og positive heltall som gir den minste tiden det er mulig å løse den aktuelle oppgaven på (se PDF til venstre).

Vi ønsker nå å finne den minimale gjennomføringstiden til en verden.

Du skal bruke Dijkstras algoritme til å finne den raskeste mulige måten å komme seg til skattkammeret H, i eksempelverdenen over. Spilleren begynner i rom A.

List opp alle estimatene til skattkammeret underveis i beregningen.

Her skal du altså gi en liste der første element er ∞ og siste element er den korrekte avstanden til skattkammeret.

2i

2(i) Ukomprimerbare verdener

Dijkstras algoritme bruker, som kjent fra kurset, $O(|E|log(|V|))$ (eventuelt $O(|V|^2)$) tid.

I den siste deloppgaven skal vi se på en spesiell type verden der vi kan regne ut den minimale gjennomføringstiden i lineær tid.

En **ukomprimerbar** verden er en verden slik at hvis vi representerer den som en graf $G = \langle V, E \rangle$, og deretter regner ut komponentgrafen C, får vi at antall noder i G vil være det samme som antall noder i C.

Dette vil være tilfelle hvis alle noder er "alene" i sin sterkt sammenhengende komponent.

Anta at vi blir gitt en ukomprimerbar verden med nøyaktig ett skattkammer og nøyaktig ett startrom og med minimale tider for alle oppgaver.

Forklar kort hvorfor det er mulig å finne den minimale gjennomføringstiden fra startrommet til skattkammeret i lineær ($O(|V| + |E|)$) tid.

Her er det ikke meningen at du skal implementere en algoritme, vi er kun ute etter en kort begrunnelse for hvorfor det er mulig.