Øvingsforelesning 10

TDT4120 - Algoritmer og datastrukturer

Øving 9

Disjunkte mengder

Oppgave 2: Hvilke påstander stemmer om skog-implementasjonen av disjunkte mengder?

- Uten stikomprimeringsheuristikken vil ikke FIND-SET(x) finne riktig respresentant.
- Etter FIND-SET(x) vil alle noder i treet som x tilhører ha samme forelder $x.p$.
- Rangen $u.rank$ til en node u er en øvre grense for høyden til u.
- Rangen $u.rank$ til en node u er nøyaktig lik høyden til u.

```
FIND-SET(x)
if x ≠ x.p
    x.p = FIND-SET(x.p)
return x.p
```

```
LINK(x, y)
if x.rank > y.rank
    y.p = x
else x.p = y
    if x.rank == y.rank
        y.rank = y.rank + 1
```

Disjunkte mengder

Oppgave 2: Hvilke påstander stemmer om skog-implementasjonen av disjunkte mengder?

- ~~Uten stikomprimeringsheuristikken vil ikke FIND-SET(x) finne riktig respresentant.~~
- Etter FIND-SET(x) vil alle noder i treet som x tilhører ha samme forelder $x.p$.
- Rangen $u.rank$ til en node u er en øvre grense for høyden til u.
- Rangen $u.rank$ til en node u er nøyaktig lik høyden til u.

```
FIND-SET(x)
if x ≠ x.p
    x.p = FIND-SET(x.p)
return x.p
```

```
LINK(x, y)
if x.rank > y.rank
    y.p = x
else x.p = y
    if x.rank == y.rank
        y.rank = y.rank + 1
```

Disjunkte mengder

Oppgave 2: Hvilke påstander stemmer om skog-implementasjonen av disjunkte mengder?

- ~~Uten stikomprimeringsheuristikken vil ikke FIND-SET(x) finne riktig respresentant.~~
- ~~Etter FIND-SET(x) vil alle noder i treet som x tilhører ha samme forelder $x.p$.~~
- Rangen $u.rank$ til en node u er en øvre grense for høyden til u.
- Rangen $u.rank$ til en node u er nøyaktig lik høyden til u.

```
FIND-SET(x)
if x ≠ x.p
    x.p = FIND-SET(x.p)
return x.p
```

```
LINK(x, y)
if x.rank > y.rank
    y.p = x
else x.p = y
    if x.rank == y.rank
        y.rank = y.rank + 1
```

Disjunkte mengder

Oppgave 2: Hvilke påstander stemmer om skog-implementasjonen av disjunkte mengder?

- ~~Uten stikomprimeringsheuristikken vil ikke FIND-SET(x) finne riktig respresentant.~~
- ~~Etter FIND-SET(x) vil alle noder i treet som x tilhører ha samme forelder $x.p$.~~
- Rangen $u.rank$ til en node u er en øvre grense for høyden til u.
- ~~Rangen $u.rank$ til en node u er nøyaktig lik høyden til u.~~

```
FIND-SET(x)
if x ≠ x.p
    x.p = FIND-SET(x.p)
return x.p
```

```
LINK(x, y)
if x.rank > y.rank
    y.p = x
else x.p = y
    if x.rank == y.rank
        y.rank = y.rank + 1
```

Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Anta at T_1 og T_2 er to ulike minimale spenntrær i en slik graf.

Minimale spenntrær - Antall unike

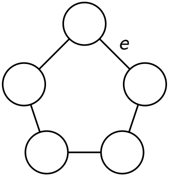
Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Anta at T_1 og T_2 er to ulike minimale spenntrær i en slik graf. La e være den kanten med lavest vekt som finnes i kun en av disse.

Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

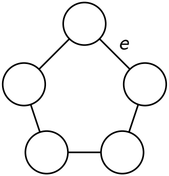
Anta at T_1 og T_2 er to ulike minimale spenntrær i en slik graf. La e være den kanten med lavest vekt som finnes i kun en av disse.

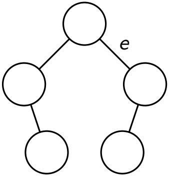


Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Anta at T_1 og T_2 er to ulike minimale spenntrær i en slik graf. La e være den kanten med lavest vekt som finnes i kun en av disse.

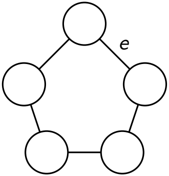


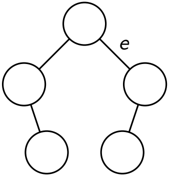


Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Anta at T_1 og T_2 er to ulike minimale spenntrær i en slik graf. La e være den kanten med lavest vekt som finnes i kun en av disse.





En slik e kan ikke eksistere.

Minimale spenntrær - Antall unike

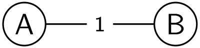
Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

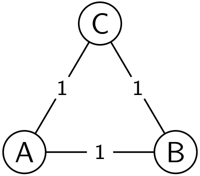
Oppgave 4: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har samme vekt?

Minimale spenntrær - Antall unike

Oppgave 3: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har forskjellige vekter?

Oppgave 4: Hvor mange ulike minimale spenntrær kan det finnes hvis alle kantene har samme vekt?





Minimale spenntrær - Kantvekter

Oppgave 5: Hvis en kant har mindre kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?

Minimale spenntrær - Kantvekter

Oppgave 5: Hvis en kant har mindre kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?

Ja, hvis ikke kan vi bytte ut en kant i spenntreet og få lavere vekt.

Minimale spenntrær - Kantvekter

Oppgave 5: Hvis en kant har mindre kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?

Ja, hvis ikke kan vi bytte ut en kant i spenntreet og få lavere vekt.

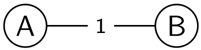
Oppgave 6: Hvis en kant har større kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?

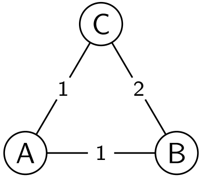
Minimale spenntrær - Kantvekter

Oppgave 5: Hvis en kant har mindre kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?

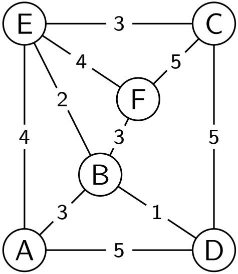
Ja, hvis ikke kan vi bytte ut en kant i spenntreet og få lavere vekt.

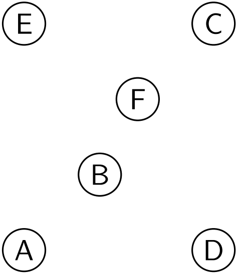
Oppgave 6: Hvis en kant har større kantvekt enn alle andre kanter i en graf, vil den være med i et minimalt spenntre?





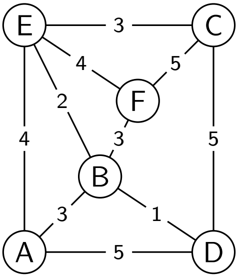
Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?

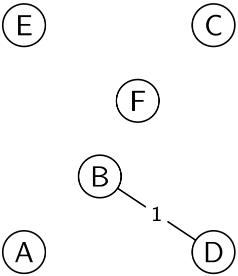




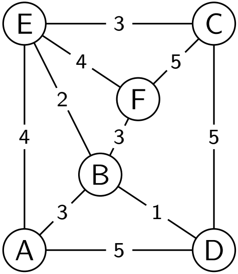
Minimale spenntrær

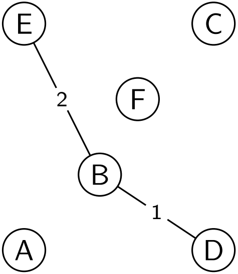
Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?





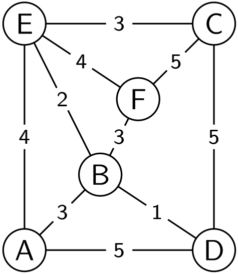
Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?

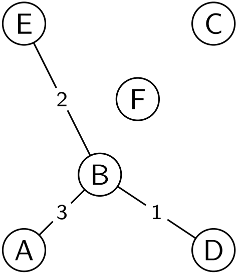




Minimale spenntrær

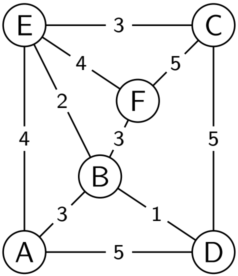
Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?

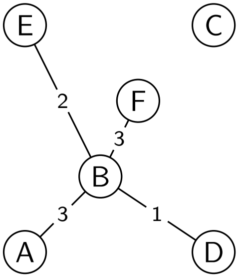




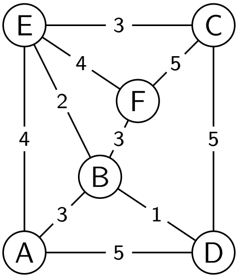
Minimale spenntrær

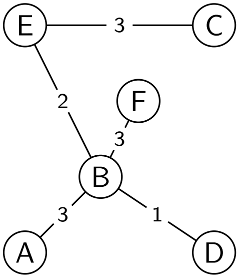
Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?





Oppgave 7: Hva er vekten av et minimal spenntre i denne grafen?





Høyere utdanning - Programmering

Oppgave 8: Ønsker en datastruktur hvor utdanningsinstitusjoner kan slås sammen, samt at man for hver institusjon kan finne den institusjonen som den for øyeblikket tilhører. Alt baserer seg på navnene til institusjonenen.

Klyngeanalyse - Programmering

Oppgave 9: Implementer en funksjon for Hamming-avstand, samt en metode for å gruppere nodene i en graf slik at man maksimerer avstanden mellom gruppene.

Klyngeanalyse - Programmering

Oppgave 9: Implementer en funksjon for Hamming-avstand, samt en metode for å gruppere nodene i en graf slik at man maksimerer avstanden mellom gruppene.

Grupperingen kan gjøres med $\text{MST-}\textsc{Kruskal}$ ved å stoppe etter $n - k$ iterasjoner.

Spenntrær - Kantrekkefølge

Oppgave 10: Hva produserer MST-PRIM hvis vi bytter min-prioritetskøen med en maks-prioritetskø?

Oppgave 11: Hva produserer MST-KRUSKAL hvis vi sorterer kantene i synkende rekkefølge?

Spenntrær - Kantrekkefølge

Oppgave 10: Hva produserer MST-PRIM hvis vi bytter min-prioritetskøen med en maks-prioritetskø?

Oppgave 11: Hva produserer MST-KRUSKAL hvis vi sorterer kantene i synkende rekkefølge?

I begge algoritmene ville vi fått samme resultat ved å sette

$$w'(u, v) = -w(u, v)$$

Spenntrær - Kantrekkefølge

Oppgave 10: Hva produserer MST-PRIM hvis vi bytter min-prioritetskøen med en maks-prioritetskø?

Oppgave 11: Hva produserer MST-KRUSKAL hvis vi sorterer kantene i synkende rekkefølge?

I begge algoritmene ville vi fått samme resultat ved å sette

$$w'(u,v) = -w(u,v)$$

Produserer et maksimalt spenntre.

Spenntrær og sykler

Oppgave 12: Hvordan kan du modifisere MST-Kruskal til å produsere en sammenhengende graf med minst én sykel og så lav sum av kantvekter som mulig?

Spenntrær og sykler

Oppgave 12: Hvordan kan du modifisere MST-Kruskal til å produsere en sammenhengende graf med minst én sykel og så lav sum av kantvekter som mulig?

Vil alltid kun ha akkurat én sykel.

Spenntrær og sykler

Oppgave 12: Hvordan kan du modifisere MST-KRUSKAL til å produsere en sammenhengende graf med minst én sykel og så lav sum av kantvekter som mulig?

Vil alltid kun ha akkurat én sykel.

Løsning: Legg til den gjenstående kanten med minst vekt.

Oppgave 13: Implementer en funksjon som gitt en liste med variabler og en liste med likheter og ulikheter, sjekker om det er mulig å tilfredsstille alle likhetene og ulikhetene samtidig.

Oppgave 13: Implementer en funksjon som gitt en liste med variabler og en liste med likheter og ulikheter, sjekker om det er mulig å tilfredsstille alle likhetene og ulikhetene samtidig.

1. Bruk disjunkte mengder til å kombinere alle variabler som skal være like.
2. Lag en graf med en node for hver disjunkte mengde fra 1.
3. For hver ulikhet $a < b$ la det gå en kant fra noden til a til noden til b.
4. Bruk DFS til å sjekke for sykler.

Oppgave 14: Implementer en funksjon som gitt en et $m \times n$-rutenett som inneholder et sett med nettstasjoner finner minste lengde på kraftledninger som må legges mellom nettstasjonene for at alle skal være koblet sammen.

Oppgave 14: Implementer en funksjon som gitt en et $m \times n$-rutenett som inneholder et sett med nettstasjoner finner minste lengde på kraftledninger som må legges mellom nettstasjonene for at alle skal være koblet sammen.

1. Bruk BFS som starter i alle nodene samtidig til å lage en liste over avstander mellom stasjonene.
2. Utfør $\mathrm{MST\text{-}K{\scriptsize RUSKAL}}$ på denne listen av kanter.

Spenntre - Splitt-og-hersk

Oppgave 15: Hvis vi splitter opp et minimalt spenntre i to deler ved å fjerne en kant er hver av de to delene et spenntre for sin del av grafen. Er disse minimale spenntrær for delene sine av grafen?

Oppgave 16: Bevis at svaret i oppgave 15 er riktig?

Spenntre - Splitt-og-hersk

Oppgave 15: Hvis vi splitter opp et minimalt spenntre i to deler ved å fjerne en kant er hver av de to delene et spenntre for sin del av grafen. Er disse minimale spenntrær for delene sine av grafen?

Oppgave 16: Bevis at svaret i oppgave 15 er riktig?

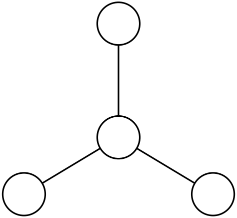
Hvis ikke, kunne vi byttet de ut med minimale spenntrær for delene sine og fått et spenntre for hele grafen med lavere vekt.

Spenntre - Splitt-og-hersk

Oppgave 17: Vil følgende splitt-og-hersk algoritme alltid finne et minimalt spenntre i en sammenhengende graf?

```
MST-D&C(G, V, w)
if |V| = 1
    return ∅
divide V into two sets V1 and V2 as evenly as possible
T1 = MST-D&C(G, V1, w)
T2 = MST-D&C(G, V2, w)
min-value = ∞
for each u ∈ V1
    for each v ∈ G.adj[u]
        if v ∈ V2 and w(u, v) < min-value
            min-edge = (u, v)
            min-value = w(u, v)
return T1 ∪ T2 ∪ {min-edge}
```

Oppgave 17: Vil følgende splitt-og-hersk algoritme alltid finne et minimalt spenntre i en sammenhengende graf?



Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

MST-D&C(G, V, w)

if $|V| = 1$
    return $\emptyset$
divide V into two sets V_1 and V_2 as evenly as possible
$T_1 = \text{MST-D\&C}(G, V_1, w)$
$T_2 = \text{MST-D\&C}(G, V_2, w)$
min-value $= \infty$
for each $u \in V_1$
    for each $v \in G.adj[u]$
        if $v \in V_2$ and $w(u, v) <$ *min-value*
            min-edge $= (u, v)$
            min-value $= w(u, v)$
return $T_1 \cup T_2 \cup \{$*min-edge*$\}$

Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

```
MST-D&C(G, V, w)
if |V| = 1
    return ∅
divide V into two sets V1 and V2 as evenly as possible   O(V)
T1 = MST-D&C(G, V1, w)
T2 = MST-D&C(G, V2, w)
min-value = ∞
for each u ∈ V1
    for each v ∈ G.adj[u]
        if v ∈ V2 and w(u, v) < min-value
            min-edge = (u, v)
            min-value = w(u, v)
return T1 ∪ T2 ∪ {min-edge}
```

Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

```
MST-D&C(G, V, w)
if |V| = 1
    return ∅
divide V into two sets V1 and V2 as evenly as possible    O(V)
T1 = MST-D&C(G, V1, w)
T2 = MST-D&C(G, V2, w)
min-value = ∞
for each u ∈ V1
    for each v ∈ G.adj[u]                                  O(E)
        if v ∈ V2 and w(u, v) < min-value
            min-edge = (u, v)
            min-value = w(u, v)
return T1 ∪ T2 ∪ {min-edge}
```

Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

```
MST-D&C(G, V, w)
if |V| = 1
    return ∅
divide V into two sets V1 and V2 as evenly as possible   O(V)
T1 = MST-D&C(G, V1, w)
T2 = MST-D&C(G, V2, w)
min-value = ∞
for each u ∈ V1
    for each v ∈ G.adj[u]                                  O(E)
        if v ∈ V2 and w(u, v) < min-value                  O(1)
            min-edge = (u, v)
            min-value = w(u, v)
return T1 ∪ T2 ∪ {min-edge}
```

Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

$\text{MST-D\&C}(G, V, w)$

if $|V| = 1$
    return $\emptyset$
divide V into two sets V_1 and V_2 as evenly as possible $\quad \mathrm{O(V)}$
$T_1 = \text{MST-D\&C}(G, V_1, w)$
$T_2 = \text{MST-D\&C}(G, V_2, w)$
min-value $= \infty$
for each $u \in V_1$
    for each $v \in G.adj[u]$ $\quad \mathrm{O(E)}$
        if $v \in V_2$ and $w(u, v) <$ *min-value* $\quad \mathrm{O(1)}$
            min-edge $= (u, v)$
            min-value $= w(u, v)$
return $T_1 \cup T_2 \cup \{$*min-edge*$\}$ $\quad \mathrm{O(V)}$

Spenntre - Splitt-og-hersk

Oppgave 18: Hva blir kjøretiden til algoritmen?

```
MST-D&C(G, V, w)
if |V| = 1
    return ∅
divide V into two sets V1 and V2 as evenly as possible    O(V)
T1 = MST-D&C(G, V1, w)
T2 = MST-D&C(G, V2, w)
min-value = ∞
for each u ∈ V1
    for each v ∈ G.adj[u]                                  O(E)
        if v ∈ V2 and w(u, v) < min-value                  O(1)
            min-edge = (u, v)
            min-value = w(u, v)
return T1 ∪ T2 ∪ {min-edge}                                O(V)
```

$$\mathrm{O}(\lg \mathrm{V}(\mathrm{E} + \mathrm{V})) = \mathrm{O}(\mathrm{E} \lg \mathrm{V})$$