Øving 10, teori: Korteste vei, én til alle

Denne øvingen fokuserer på én til alle.

Enkelte av oppgavene er *basert* på *tidligere eksamensoppgaver*. Denne øvingen kan derfor gi en liten pekepinn på hvordan man ligger an til en eventuell eksamen.

Your answer passed the tests! Your score is 100.0% ×

Question 1: Korteste vei, én til alle 1

×

Hva går korteste vei-problemet ut på?

- ( ) Finne sykler som minimerer summen av kantvektene
- ( ) Finne stier fra noder til andre noder med færrest mulig kanter
- (•) Finne stier fra noder til andre noder som minimerer summen av kantvektene
- ( ) Finne sykler med færrest mulig kanter

Question 2: Korteste vei, én til alle 2

×

Hvilke(n) påstand(er) er korrekt(e)?

- [ ] Det finnes aldri mer enn én korteste vei mellom 2 noder.
- [x] Ved å snu alle kantene i en graf kan man finne korteste alle-til-én veier, gitt at man vet hvordan man finner korteste en-til-alle veier.
- [ ] Korteste vei har ikke optimal substruktur.
- [ ] DFS vil finne korteste vei til alle noder i en graf
- [x] Selv om man har korteste vei fra node A til B og korteste vei fra node B til C vil man ikke kunne garantere at A -> B -> C er korteste vei mellom A og C.

Question 3: Korteste vei, én til alle 3

×

Anta at du har korteste vei mellom nodene A og C, som går gjennom node B. Hvilken påstand er korrekt?

- ( ) Ingen av de andre påstandene er korrekt
- ( ) Vi har også korteste veg mellom B og C, men ikke A og B
- ( ) Vi har også korteste veg mellom A og B, men ikke B og C
- (•) Vi har også korteste veg mellom A og B, og B og C

Question 4: Korteste vei, én til alle 4

×

Hvilken påstand er korrekt?

- ( ) Det er umulig å detektere om en graf har negative sykler.
- (•) Bellman-Ford algortimen finner korteste vei med negative kanter, men ikke negative sykler
- ( ) Dijkstra's algoritme finner korteste veg med negative kanter, men ikke sykler
- ( ) Dijkstra's algoritme finner korteste veg med både negative kanter og negative sykler
- ( ) Bellman-Ford algortimen finner korteste vei med både negative kanter og negative sykler
- ( ) Vi kan ikke fjerne 0-vektede sykler fra en sti for å lage en en ny sti med samme vekt.

Question 5: Korteste vei, én til alle 5

×

Vi har $u.d = 5$, $v.d = 10$, $w(u, v) = 2$. Hva blir $u.d$ og $v.d$ etter $RELAX(u, v, w)$?

- ( ) $u.d = 5, v.d = 10$
- (•) $u.d = 5, v.d = 7$
- ( ) $u.d = 3, v.d = 10$
- ( ) $u.d = 10, v.d = 5$
- ( ) $u.d = 5, v.d = 5$
- ( ) $u.d = 3, v.d = 7$
- ( ) $u.d = 3, v.d = 8$
- ( ) $u.d = 5, v.d = 8$

Question 6: Korteste vei, én til alle 6

×

Vi har $u.d = 6$, $v.d = 9$, $w(u, v) = 4$. Hva blir $u.d$ og $v.d$ etter $RELAX(u, v, w)$?

- (•) $u.d = 6, v.d = 9$

- $u.d = 3, v.d = 9$
- $u.d = 6, v.d = 5$
- $u.d = 4, v.d = 3$
- $u.d = 1, v.d = 9$

Question 7: Korteste vei, én til alle 7

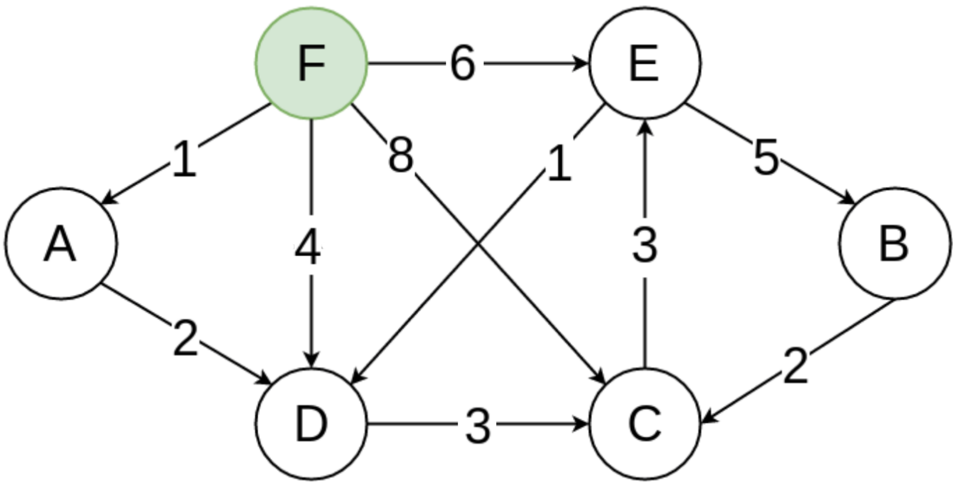
×

Anta at du har kjørt en av pensumalgoritmene som løser korteste-vei problemet, én til alle, for grafen $G = (V, E)$, med startnode s. Dersom en node $v \in V - s$ har $v.\pi = NIL$, hva er $v.d$?

- ∞
- NIL
- 0
- $\sum_{u \in P} u.d$, hvor P er den korteste stien mellom s og v

Question 8: Bellman-Ford 1

×



I denne oppgaven skal du bruke Bellman-Ford (se side 651) for å finne korteste vei fra node F.

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter kallet til INITIALIZE-SINGLE-SOURCE i algoritmen - hva er *A.d*?

- π
- 0
- ∞

- [ ] 1

Question 9: Bellman-Ford 2

(Bruk grafen fra Bellman-Ford 1)

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter kallet til INITIALIZE-SINGLE-SOURCE i algoritmen - hva er *F.d*?

- [ ] π
- [x] 0
- [ ] 1
- [ ] ∞

Question 10: Bellman-Ford 3

(Bruk grafen fra oppgave Bellman-Ford 1)

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter én iterasjon av for-løkken på linje 2-4 - hva er *D.d*?

- [ ] 0
- [ ] 3
- [x] 4
- [ ] ∞

Question 11: Bellman-Ford 4

(Bruk grafen fra oppgave Bellman-Ford 1)

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter én iterasjon av for-løkken på linje 2-4 - hva er *B.d*?

- ( ) 0
- ( ) 11
- ( ) 5
- (x) ∞

Question 12: Bellman-Ford 5

×

(Bruk grafen fra oppgave Bellman-Ford 1)

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter 2 iterasjon av for-løkken på linje 2-4 - hva er *E.d*?

- (x) 6
- ( ) 4
- ( ) 5
- ( ) ∞

Question 13: Bellman-Ford 6

×

(Bruk grafen fra oppgave Bellman-Ford 1)

Merk: Anta at algoritmen slakker utgående kanter fra noder i leksikalsk rekkefølge. Altså, utgående kanter fra node A før node b, utgående kanter fra node B før node C, osv.

Etter 2 iterasjon av for-løkken på linje 2-4 - hva er *C.d*?

- ( ) 8
- ( ) 9
- (x) 6
- ( ) 7

Question 14: Bellman-Ford 7

×

Hva er kjøretiden til Bellman-Ford?

- ( ) $O(V \log E)$

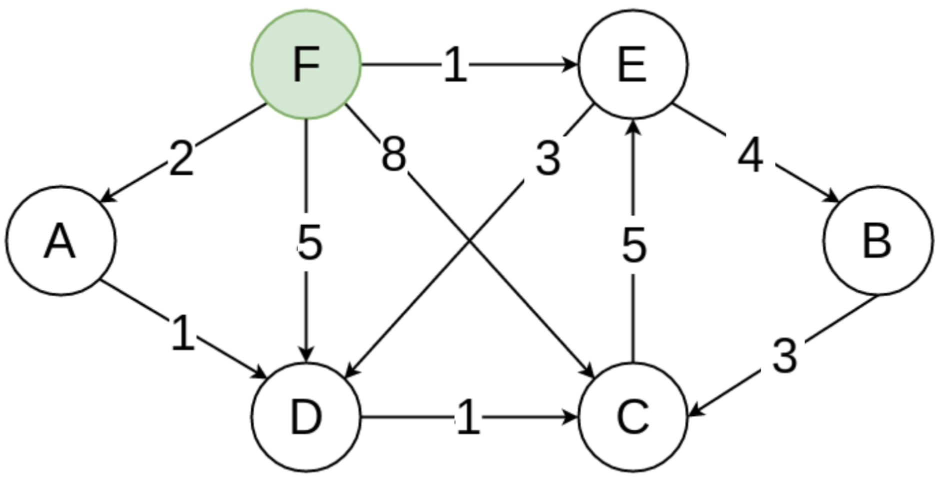
- ( ) $O(V^2)$
- ( ) $O(E \log V)$
- (•) $O(VE)$
- ( ) $\Omega(E^2)$

Question 15: DAG shortest path 1

Hvilke(n) påstand(er) er korrekt(e)?

- [ ] Vi kan bruke DAG shortest path på grafen i Bellman-Ford-oppgaven over
- [x] Ingen av de andre påstandene er korrekte
- [ ] Vi kan ikke bruke DAG shortest path med negative kanter
- [ ] Kjøretiden til DAG shortest path er lik $\Theta(VE)$

Question 16: Dijkstras algoritme 1



I denne oppgaven skal du bruke grafen over. Vi skal finne korteste vei fra node F. For dette skal vi bruke Dijkstra (se side 658).

Hvis du kan velge mellom to noder velger du i alfabetisk rekkefølge

I hvilken rekkefølge legges nodene til i *S*?

- ( ) FEADBC
- (•) FEADCB
- ( ) FABCDE
- ( ) FEDCBA
- ( ) FEDCAB

Question 17: Dijkstras algoritme 2

×

(Bruk grafen fra oppgave Dijkstras Algoritme 1)

Hva er C.d etter én interasjon av while løkken i linje 4-8?

- ○ 4
- ◉ 8
- ○ 5
- ○ ∞

Question 18: Dijkstras algoritme 3

×

(Bruk grafen fra oppgave Dijkstras Algoritme 1)

Hva er D.d etter to iterasjon av while-løkken på linje 4-8?

- ○ ∞
- ○ 3
- ○ 5
- ◉ 4

Question 19: Dijkstras algoritme 4

×

(Bruk grafen fra oppgave Dijkstras Algoritme 1)

Hva er C.d etter 3 iterasjoner av while-løkken i linje 4-8?

- ◉ 8
- ○ 6
- ○ 7
- ○ 5
- ○ 4

Question 20: Dijkstras algoritme 5

×

Hva er C.d etter 4 iterasjoner av while-løkken i linje 4-8?

- ○ 8
- ○ 7
- ◉ 4
- ○ 6
- ○ 5

Question 21: Dijkstras algoritme 6

×

Hva er B.d etter 6 iterasjoner av while-løkken i linje 4-8?

- ○ 7
- ◉ 5
- ○ 6
- ○ 8

Question 22: Dijkstras algoritme 7

×

Hvilke(n) påstand(er) er korrekt(e)?

- [ ] Ingen av de andre påstandene er korrekte
- [ ] Å øke alle kantvekter med en konstant k slik at alle kantvekter blir positive vil gjøre at Dijkstra fungerer med negative kanter.
- [x] Dijkstra er en grådig algoritme.

Question 23: Dijkstras algoritme 8

×

Hvordan beviser Teorem 24.6 (side 659) Dijkstra sin korrekthet?

- ○ Viser at avstandsestimatet til nodene alltid går nedover
- ○ Viser at vi kjører RELAX i rekkefølge over korteste sti
- ◉ Viser at neste node som velges til enhver tid har riktig avstandsestimat.

- Viser at vi kjører RELAX V-1 ganger på hver kant

Submit

>_