

- | | | | | |
|--|--|--|--|--|
| | | | | |
|--|--|--|--|--|
- ☒ Matisen i c) er symmertisk fordi G er urettet
 - ☒ c) er en nabomatise for figuren i a)
 - ☒ a) er en urettet graf
 - ☒ b) er en naboliste-representasjon for figuren i a)
 - ☒ $|V| = 6$ og $|E| = 6$
 - ☒ Graden til node 3 er 3

Question 2: Representasjon av grafer

Oppgaven ble godkjent!



Gitt at $|V| = |E|^2$, hvordan kan grafen G lagres mest plasseffektivt i denne situasjonen?

- ☐ En lenket liste
- ☒ Nabolister
- ☐ En nabomatrise
- ☐ Nabolister og nabomatrise er like plasseffektivt
- ☐ Nabolister, nabomatrise og lenket liste er like plasseffektivt

Question 3: Representasjon av grafer

Oppgaven ble godkjent!



Anta at du har to noder, $u, v \in V$. Hvor lang tid vil det ta å sjekke om det finnes en kant, $e \in E$, som går fra u til v gitt at grafen G er representert ved hjelp en nabomatrise? Anta at du ikke vet noe om hvor mange kanter eller hvor mange noder det finnes i G .

- ☒ $O(1)$
- ☐ $O(|E| + |V|)$
- ☐ $O(|E|)$
- ☐ $O(|V|)$

Question 4: Representasjon av grafer

Oppgaven ble godkjent!



Hvor lang tid vil tilsvarende oppslag ta hvis G er en naboliste-representasjon og det går minst én kant ut fra hver node?

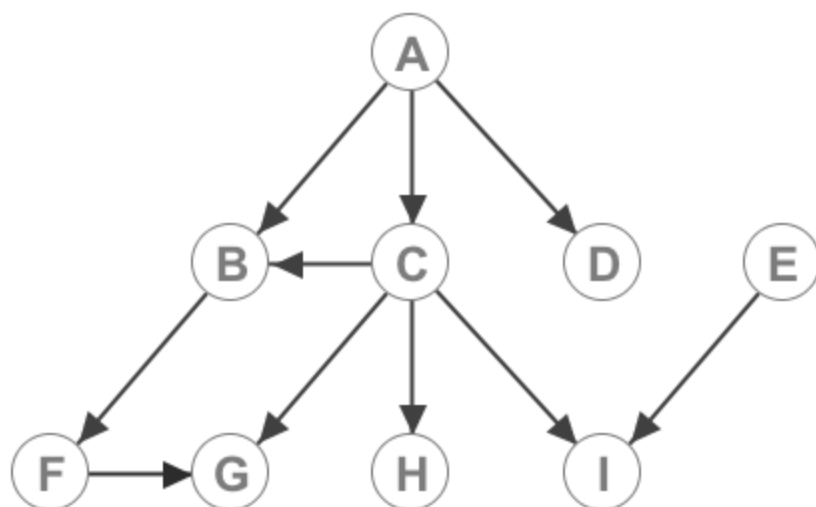
- ☒ $O(|V|)$
- ☐ $O(1)$
- ☐ $O(|E| + |V|)$
- ☐ $O(|E|)$

Question 5: Bredde-først-søk

Oppgaven ble godkjent!



BFS blir kjørt med påfølgende graf med *A* som rotnode. I hvilken rekkefølge blir de fire første nodene farget svart?



Anta at alle konflikter løses ved hjelp av leksikografisk ordning (ved eventuelle konflikter velges den noden med bokstav tidligst i alfabetet, altså *A* før *B*, *B* før *C* osv.)

- ☐ *A, B, D, C*
- ☒ *A, B, C, D*
- ☐ *A, B, F, G*
- ☐ *B, C, D, F*
- ☐ *A, C, D, B*
- ☐ *A, C, B, D*
- ☐ *B, C, D, E*

Question 6: Bredde-først-søk

Oppgaven ble godkjent!



Hvilke(n) påstand(er) stemmer om BFS?

- ☒ Implementeres vanligvis med en kø
- ☐ Ingen av påstandene stemmer
- ☐ Implementeres vanligvis rekursivt
- ☐ Implementeres vanligvis med en stakk
- ☐ Implementeres vanligvis med en heap

Question 7: Bredde-først-søk

Oppgaven ble godkjent!



For hvilket av alternativene under er vi garantert at bredde-først-søk finner korteste vei i en vilkårlig sammenhengende graf?

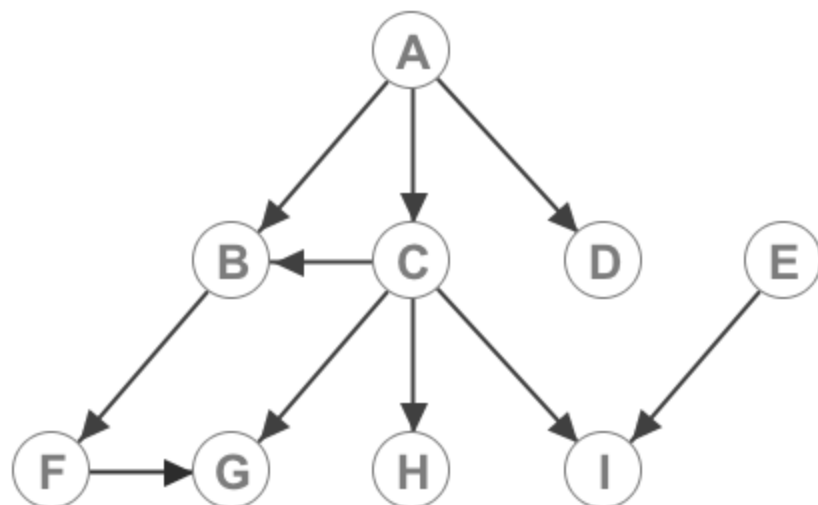
- ☐ Ingen negative kanter
- ☒ Alle kantene har lik ikke-negativ vekt
- ☐ Alle kantene har lik vekt

Question 8: Dybde-først-søk

Oppgaven ble godkjent!



DFS blir kjørt med påfølgende graf med *A* som rotnode. I hvilken rekkefølge blir de fire første nodene farget svart?



Anta at alle konflikter løses ved hjelp av leksikografisk ordning (ved eventuelle konflikter velges den noden med bokstav tidligst i alfabetet, altså *A* før *B*, *B* før *C* osv.)

- ☐ *A, C, D, B*
- ☐ *B, C, D, F*
- ☐ *A, B, C, D*
- ☐ *A, C, B, D*
- ☐ *A, B, F, G*
- ☐ *G, H, B, F*
- ☒ *G, F, B, H*
- ☐ *B, C, D, E*
- ☐ *A, B, D, G*

Question 9: Dybde-først-søk

Oppgaven ble godkjent!



Hvilke(n) påstand(er) stemmer om DFS?

- ☐ Det er svært unaturlig å implementere DFS med rekursjon
- ☒ Det er svært unaturlig å implementere DFS med kø
- ☐ Det er naturlig å implementere DFS med kø, stakk, rekursjon og heap
- ☒ Det er svært unaturlig å implementere DFS med heap
- ☐ Det er svært unaturlig å implementere DFS med stakk

Question 10: Dypde-først-søk

Oppgaven ble godkjent!



Et dypde-først-søk kan brukes til å klassifisere kantene i en graf. Hvilken av følgende kanttyper betegner en kant som går fra en forgjenger (ancestor) til en etterkommer (descendant)?

- ☐ Cross edge
- ☐ Back edge
- ☒ Tree edge

Question 11: Dybde-først-søk

Oppgaven ble godkjent!



Hva slags type kant kan vi ha kommet til når vi kommer til en node som allerede er farget svart i et dypde-først-søk?

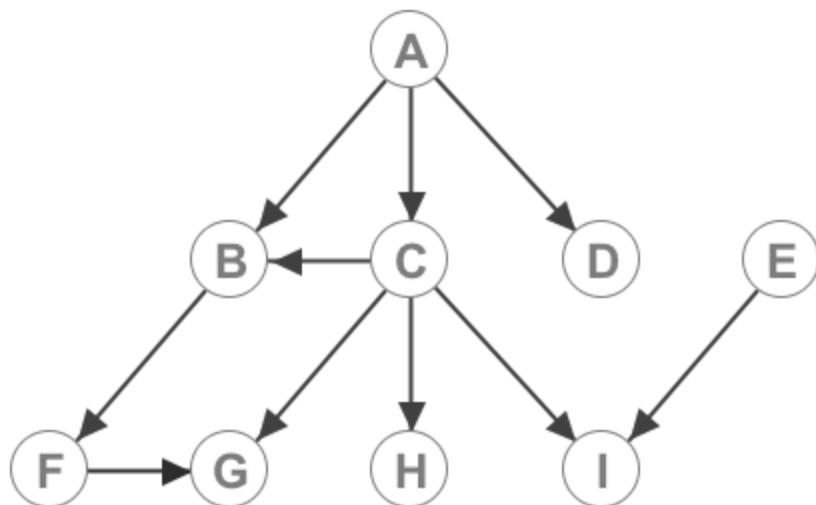
- ☒ Cross edge
- ☒ Forward edge
- ☐ Tree egde
- ☐ Back edge

Question 12: Topologisk sortering

Oppgaven ble godkjent!



Hvilke(n) av følgende alternativ er en gyldig topologisk sortering?



Hint: En graf kan ha flere mulige topologiske sorteringer. I stedet for å lage en topologisk sortering av grafen, bør du heller sjekke hvilke av alternativene som overholder kravene til en topologisk sortering

- ☐ *E, A, I, D, C, H, B, F, G*
- ☐ *E, A, D, C, I, H, B, G, F*
- ☒ *A, E, D, C, I, H, B, F, G*
- ☐ *A, B, F, G, C, H, I, D, E*

Question 13: Topologisk sortering

Oppgaven ble godkjent!



Du ønsker å lage en topologisk sortering av en graf $G = (V, E)$. Hvilke av følgende kriterier må være sanne (for grafen G) for at det skal finnes en topologisk sortering?

- ☒ Den må være rettet og asyklisk (en DAG)
- ☐ Alle kantvektene må være like
- ☐ Den må ha positive kantvekter
- ☐ Alle de andre alternative over må være riktige

Question 14: Tidligere eksamensoppgave

Oppgaven ble godkjent!



Du prøver å implementere BFS for urettede grafer, men på grunn av en kodefeil, er rekkefølgen på nodene i køen din ikke lenger FIFO, men helt vilkårlig. Kan du nå være sikker på å besøke alle nodene?

- ☒ Ja, dersom grafen er sammenhengende
- ☐ Ja, for alle grafer
- ☐ Nei
- ☐ Ja, dersom grafen er en skog

Question 15: Best-case-kjøretid for BFS og DFS

Oppgaven ble godkjent!



Hva er best-case-kjøretid for BFS og DFS gitt implementasjonen i læreboken?

- ☒ $O(1)$ for BFS og $O(|V| + |E|)$ for DFS
- ☐ $O(1)$ for begge
- ☐ $O(|V| + |E|)$ for begge
- ☐ $O(|V| + |E|)$ for BFS og $O(1)$ for DFS