

**Eksamensoppgave i TDT4120 Algoritmer og datastrukturer**

Faglig kontakt under eksamen

Magnus Lie Hetland

Tlf.

91851949

Eksamensdato

15. august 2013

Eksamenstid (fra-til)

0900–1300

Hjelpemiddelkode

D. Ingen trykte eller håndskrevne hjelpemidler  
tillatt. Bestemt, enkel kalkulator tillatt.

Målform/språk

Bokmål

Antall sider

5

Antall sider vedlegg

0

Kontrollert av

Ole Edsberg

---

  
Dato

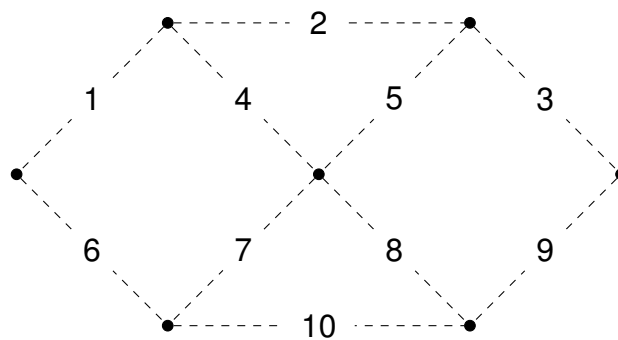
Sign

- ! **Les alle oppgavene før du begynner, disponer tiden og forbered spørsmål til faglærer ankommer lokalet.** Gjør antagelser der det er nødvendig. Skriv kort og konsist på **angitt sted**. Lange forklaringer og utledninger som ikke direkte besvarer oppgaven tillegges liten eller ingen vekt.

Algoritmer kan beskrives med tekst, pseudokode eller programkode, etter eget ønske (med mindre annet er oppgitt), så lenge det klart fremgår hvordan den beskrevne algoritmen fungerer. Korte, abstrakte forklaringer kan være vel så gode som utførlig pseudokode, så lenge de er presise nok. Algoritmer som konstrueres bør generelt være så effektive som mulig, med mindre annet er opplyst. Kjøretider oppgis med asymptotisk notasjon, så presist som mulig. Alle deloppgavene teller like mye.

1. Betrakt den vektete, urettede grafen i svar-ruten, nedenfor. Tegn inn et minimalt spenntre for grafen. (Dvs., tegn over/«fyll inn» over de stiplede linjene med heltrukne linjer for å indikere kanter i spenntreet.)

Svar:



Spenntreet skal inneholde kantvektene 1, 2, 3, 4, 6, 8.

2. Anta at du har en komplett (urettet) graf med  $n$  noder. Du lager så en rettet, asyklisk graf ved å gi hver kant en retning (men det er ikke spesifisert hvilken retning). Hvor mange ulike topologiske sorteringer kan den resulterende rettede grafen ha? Forklar svært kort.

Svar: Én, fordi ordningen av ethvert par med noder er gitt.

3. Hva er worst-case-kjøretiden til Quicksort? Oppgi svaret i  $\Theta$ -notasjon.

Svar:  $\Theta(n^2)$

4. Counting Sort har bedre asymptotisk kjøretid enn Mergesort, fordi Counting Sort gjør en sentral antagelse som Mergesort ikke gjør. Hvilken antagelse er dette?

Svar: Den baserer seg på at verdiene er heltall i et begrenset verdiområde.

5. Dijkstras algoritme har bedre asymptotisk kjøretid enn Bellman-Ford, fordi Dijkstras algoritme gjør en sentral antagelse som Bellman-Ford ikke gjør. Hvilken antagelse er dette?

Svar: Den baserer seg på antagelsen at ingen av kantvektene er negative.

6. I en vanlig implementasjon, hvilken av algoritmene Heapsort og Mergesort bruker normalt mest minne?

Svar: Mergesort

7. La høyden til et perfekt balansert binærtre være lengden til (dvs., antall kanter i) enhver sti fra rota til en løvnode. Hva er høyden til et perfekt balansert binærtre med 1023 interne noder?

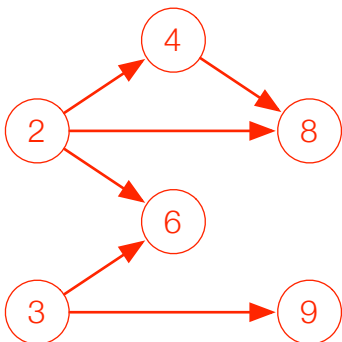
Svar: 10

8. Betrakt følgende beskrivelse av en algoritme anvendt på et veinett mellom byer.  
«Gå gjennom alle byene etter tur. For hver by B, betrakt alle andre par med byer. For hvert slikt par A, C, sjekk om det er en snarvei å reise fra A til C via B.»  
Hvilken algoritme er det som beskrives?

Svar: Floyd-Warshall

9. La  $G = (V, E)$  være en rettet graf der  $V = \{2, 3, 4, 6, 8, 9\}$  og  $E$  består av kanter  $(u, v)$  der  $v$  er delelig på  $u$ . (For eksempel vil kanten  $(2, 4)$  finnes, fordi 4 er delelig på 2.) En pensumalgoritme kjøres på grafen, og gir følgende ordning av nodene:  
2, 4, 8, 6, 3, 9  
Hvilken algoritme er det snakk om?

Svar: Her er både DFS og BFS riktig. Topologisk sortering er her ekvivalent med DFS.



10. Din venn Alan har bevist at problemet H er uløselig. Du har et annet problem K, som du også tror er uløselig. Hvordan vil du bruke problemene H og K for å vise at K er uløselig?

Svar: Reduser fra H til K.

11. Forklar svært kort hvorfor vertex covering-problemet er et spesialtilfelle av set covering-problemet.

Svar: Nabolagene fra vertex covering utgjør mengdene i set covering.

12. Beskriv med kode eller pseudokode en rekursiv algoritme for å skrive ut alle permutasjoner av en liste med unike elementer.

Svar:

def permutasjoner av liste:

  hvis lista er tom:

    returner tom permutasjon

  for hvert element x i liste:

    for hver permutasjon p av de resterende elementene:

      returner x + p

for hver permutasjon av liste:

  skriv ut permutasjon

(Her er det stor fleksibilitet i hvordan oppgaven besvares.)

13. Betrakt følgende rekurrens:

$$T(n) = 3T(n-1) - 2T(n-2) \quad \text{for } n > 1, \text{ med } T(0) = 0 \text{ og } T(1) = 1.$$

Løs rekurrensen.

**Hint 1:**  $T(n) - T(n-1) = 2(T(n-1) - T(n-2))$ .

**Hint 2:** La  $F(n) = T(n) - T(n-1)$ .

Svar: La  $F(n) = T(n) - T(n-1)$  og løs rekurrensen  $F(n) = 2F(n-1)$  for  $n > 1$ , med  $F(1) = 1$ . Vi får da at  $F(n) = T(n) - T(n-1) = 2^{n-1}$ , som gir oss  $T(n) = T(n-1) + 2^{n-1}$ , som igjen gir oss  $T(n) = 2^n - 1$ .

**Merk:** I den opprinnelige oppgaven sto det  $T(n-3)$  i stedet for  $T(n-2)$  i hint 1. Det ble opplyst om feilen på eksamen, men oppgaven tas likevel ut av sensur det er til fordel for studenten.

14. I en rettet graf, la hver sykel ha en kostnad satt til gjennomsnittet av kantvektene i sykelen. Anta at alle vektene er positive heltall. Beskriv en algoritme som finner den laveste sykelkostnaden i grafen.

**Hint 1:** Hvordan kan du finne ut om en rettet graf inneholder en negativ sykel?

**Hint 2:** Hvordan kan du finne det minste positive heltallet som, hvis du trekker det fra alle kantvektene, gir en negativ sykel i grafen?

Svar: Bruk Bellman-Ford til å avgjøre om grafen har en negativ sykel. Bruk binærsøk (mellom 0 og høyeste kantvekt) for å finne den minste subtrahenden som gir en negativ sykel. Trekk én fra denne, så har du den laveste gjennomsnittskostnaden.

**Merk:** I oppgaven ligger en antagelse om at gjennomsnittskostnaden er et heltall, noe som ikke trenger være tilfellet. På grunn av dette tas oppgaven ut av sensur der det er til fordel for studenten.

15. Du skal lage en algoritme som hjelper politiet med å sette opp veisperringer. Som input får du en urettet graf med følgende egenskaper:

- Kantene representerer veier og nodene representerer veikryss
- Hver kant har en lengde som representerer veistrekningen
- En bestemt node representerer åstedet for en forbrytelse
- Et sett med noder angir havner og grenseoverganger, såkalte «kantnoder»

Du får også opplyst maksimal hastighet forbryterne kan ha beveget seg i fra åstedet, samt hvor lang tid som har gått siden forbrytelsen ble begått. Du kan anta at forbryterne ikke kan ha nådd frem til noen av kantnodene ennå. Algoritmen skal fortelle politiet hvor (dvs. på tvers av hvilke kanter) de skal sette opp veisperringer, slik at (i) forbryterne ikke kan komme seg til en kantnode (ved å traversere kanter) uten å passere en veisperring, og (ii) antallet veisperringer er så lite som mulig.

Svar: Bruk Dijkstras algoritme til å finne ut hvilke noder forbryterne alt kan ha nådd (dvs. hvor de kan være) med gitte avstander, maks-hastighet og tid. Koble disse til en kildenode, og koble kantnodene til en sluk-node. Sett alle kant-kapasiteter til 1. Finn minimalt snitt (f.eks. vha. Edmonds-Karp) mellom kilde og sluk.