NTNU
Institutt for datateknikk
og informasjonsvitenskap

Eksamensoppgave i TDT4120 Algoritmer og datastrukturer

Faglig kontakt under eksamen	Magnus Lie Hetland
Tlf.	91851949
Eksamensdato	7. desember 2013
Eksamenstid (fra–til)	0900–1300
Hjelpemiddelkode	D. Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.
Målform/språk	Bokmål
Antall sider	6
Antall sider vedlegg	0
Kontrollert av	Pål Sætrom

_____

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. Hva er det minimale og maksimale antall kanter i en sammenhengende, urettet graf med n noder?

Minimum:
Maksimum:

2. Hva er det minimale og maksimale antall noder i et binærtre med rot og dybde n? (Dybden er det maksimale antall kanter på en sti fra rota.)

Minimum:
Maksimum:

3. Hva er forventet (*average-case*) asymptotisk kjøretid for et mislykket søk i en hash-tabell med m plasser (*slots*) som inneholder n elementer? Uttrykk svaret som funksjon av n og m. (Du kan anta enkel, uniform hashing og at kollisjoner løses vha. *chaining*.)

Svar:

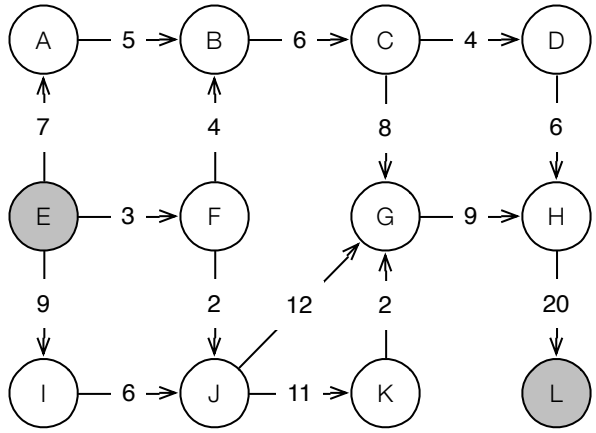
4. Et problem kan ofte brytes ned i delproblemer, der enkelte delproblemer er avhengige av andre delproblemer. For å løse problemet, må delproblemene løses i en rekkefølge som tilfredsstiller disse avhengighetene. Om man ser på delproblemene som noder og avhengighetene som kanter i en rettet graf, hva kalles en slik rekkefølge?

Svar:

5. Løs rekurrensen $T(n) = T(n+1) - 2^n$; $T(1) = 1$. Oppgi svaret eksakt.

Svar:

6. Følgende rettede graf representerer et flytnettverk, med angitte kapasiteter. Hva er maksimal flyt fra node E til node L?



Svar:

7. Hva er maks-haug-egenskapen (*the max-heap property*)?

Svar:

8. Sortering av n elementer kan generelt ikke utføres i $O(n)$ tid. Likevel klarer f.eks. tellesortering (*counting-sort*) dette. Hvordan kan det ha seg?

Svar:

9. Din venn påstår å ha oppfunnet en algoritme som løser *traveling salesman*-problemet i $O(n \log(\log(n)))$ tid, der n er antall noder i grafen. Hvis din venns påstand var korrekt, ville det få noen viktige konsekvenser? Forklar kort.

Svar:

10. Du har n venner. Du vet at noen av vennene dine hater hverandre. Hatet er alltid gjensidig. Du vet akkurat hvilke par av venner som hater hverandre. Du ønsker å «unfriende» (avslutte vennskapet med) noen av vennene, slik at ingen av de gjenværende vennene hater hverandre. Din oppgave er å avgjøre om du kan løse dette problemet ved å unfriende k venner, der k er en input-parameter. Du kan anta at dette beslutningsproblemet er i NP. Vis at det er NP-komplett. Forklar deg presist, og inkluder alle nødvendige deler av et slikt argument.

Svar:

11. Student Lurvik hevder at han har funnet en måte å få Dijkstras algoritme til å tåle negative kanter: Finn den korteste kanten, altså den «mest negative», og kall lengden dens for w. Adder $|w| + 1$ til alle kantlengdene, slik at alle kantene får positive lengder. Kjør deretter DIJKSTRA på den modifiserte grafen; ta svaret man får og bruk de tilsvarende kantene i den originale grafen. Bevis at dette alltid gir rett svar, eller finn et moteksempel som viser at Lurvik sin algoritme kan gi feil svar.

Svar:

12. Dijkstras algoritme kan implementeres ved å bruke en binærhaug (*binary heap*) som prioritetskø, eller ved å bare bruke den usorterte nodetabellen som prioritetskø. Dette fører til forskjellige kjøretider. Gitt følgende pseudokode for DIJKSTRA, oppgi de linjene som blir påvirket av dette valget og forklar hvorfor kjøretidene blir hhv. $O(E \lg V)$ og $O(V^2)$. Når bør man bruke hva?

```
DIJKSTRA(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = G.V
4  while Q ≠ ∅
5      u = EXTRACT-MIN(Q)
6      S = S ∪ {u}
7      for each vertex v ∈ G.Adj[u]
8          RELAX(u, v, w)
```

Linjer som påvirkes (linjenummer):
Forklaring av kjøretider:

Når man bør bruke usortert liste/tabell:

13. Multiplikasjon av en matrise med n rader og m kolonner med en matrise som har m rader og k kolonner har kjøretiden $O(nmk)$. Algoritmen for å løse 0/1-knapsack-problemet hvor man skal stjele n gjenstander ved hjelp av en sekk som har kapasitet k kjører på $O(nk)$. Førstnevnte algoritme har polynomisk kjøretid, mens den andre algoritmen har eksponentiell kjøretid. Hvordan forklarer man dette?

Svar:

14. Din venn Smartnes mener han har lest i pensum at worst-case-kjøretiden til sammenligningsbasert sortering er $\Omega(n \lg n)$, men han skjønner ikke helt hva det betyr. Forklar kort hva det betyr, eller forklar kort hvorfor dette ikke gir mening.

Svar:

15. To personer skal besøke en sekvens med byer. Du har oppgitt en ordnet sekvens av n byer, og avstandene mellom alle par med byer. Du skal fordele byene i to subsekvenser (der byene ligger i sin opprinnelige rekkefølge, men ikke nødvendigvis rett etter hverandre i den opprinnelige sekvensen) sånn at person A besøker byene i den første sekvensen (i rekkefølge) men ingen av de andre, person B besøker byene i den andre sekvensen (i rekkefølge) men ingen av de andre, og sånn at summen av avstandene A og B reiser totalt er minimert. (Hver by skal altså besøkes av nøyaktig én av A eller B.) Anta at A og B starter i de første byene i sine respektive subsekvenser. Beskriv en effektiv algoritme som løser problemet. Hva blir kjøretiden?

Svar: