

TDT4120 Algoritmer og datastrukturer

Eksamen, 5. august 2025, 09:00–13:00

Faglig kontakt Magnus Lie Hetland

Hjelpemiddelkode E

Oppgaver

- 1 Sammenlign lenkede lister (*linked lists*) med tabeller (*arrays*). Hvilke fordeler og ulemper har de, sammenlignet med hverandre? Forklar.
- 2 Kan man ha en ustabil matching om man bare har to par? Forklar kort.
- 3 Hvis et flytnett (*flow network*) kun har heltallskapasiteter, så vil den største mulige flytverdien også være heltallig. Om vi finner flyten med FORD-FULKERSON-metoden, vil flyten i hver eneste kant også være et heltall. Kan det finnes andre optimale løsninger der flyten i noen av kantene *ikke* er heltall? Forklar.
- 4 Om du bygger et binært søketre av n elementer, hva blir kjøretiden til søk i treet etterpå, i beste og verste tilfelle, og i gjennomsnitt? Forklar.

Du bruker altså TREE-INSERT for å sette inn ett og ett element, der treet til å begynne med er tomt, og vi er ute etter kjøretiden til TREE-SEARCH på det ferdige treet (*best-case*, *worst-case* og *average-case*). Du kan anta at elementene er tallene $1 \dots n$ i en (uniformt) tilfeldig rekkefølge.

Algoritme 1

```
TOO-TIRED-FOR-THIS( $X, n, k$ )
1  for  $i = 2$  to  $n$ 
2       $y = X[i]$ 
3       $j = i - 1$ 
4      while  $j > 0$  and  $X[j] > y$  and  $j \geq i - k$ 
5           $X[j + 1] = X[j]$ 
6           $j = j - 1$ 
7       $X[j + 1] = y$ 
8      if  $j > 0$  and  $X[j] > y$ 
9          return FALSE
10 return TRUE
```

- 5 Se prosedyren i algoritme 1. Hva er kjøretiden i beste og verste tilfelle? Hva kan prosedyren brukes til? Forklar.
- 6 Løs den algoritmiske rekurrensen $T(n) = 4T(9n/16)/3 + \sqrt{n}$. Forklar.
- 7 Din venn Gløgsund spiller et spill, der hun utforsker en verden, og skal innom et sett med landsbyer. Hun ønsker å utforske alle landsbyene så effektivt som mulig, og mener det blir omtrent som handelsreiseproblemet (*the traveling-salesperson problem*, TSP).

Hun er usikker på om det er *fullt* så vanskelig, riktignok, av to grunner:

1. Det er snakk om et spesialtilfelle, der noder u, v er punkter i planet, og kostnaden $c(u, v)$ er avstanden i rett linje mellom dem.
2. I TSP er det forbudt å besøke en node flere ganger, men her kan hun jo gjøre som hun vil.

Hun spør sin venn Klokland, som også er litt usikker. Hun kan fortelle at det første punktet nok ikke hjelper; det betyr bare at det er snakk om såkalt *euklidsk* TSP, som fortsatt er NP-hardt. Men det andre punktet er hun ikke sikker på.

Hjelp Gløgsund, ved å enten finne en effektiv algoritme for problemet, eller ved å bruke en reduksjon til å vise at det er NP-hardt. Forklar svaret ditt.

- 8 (*Forts. fra oppg. 7*) Klokland oppdager at Gløgsund ikke har lest instruksjonene ordentlig: Spillet støtter teleportasjon (*fast-travel*), så hun helt uten kostnad kan forflytte seg til landsbyer hun allerede har vært innom. Det gjør jo at hun kan bruke mindre tid på å gjøre seg ferdig, men begge er fortsatt usikre på hvor vanskelig det er å finne den *optimale* reiseruten.

Hjelp Gløgsund, ved å enten finne en effektiv algoritme for problemet, eller ved å bruke en reduksjon til å vise at det er NP-hardt. Forklar svaret ditt.

- 9 Dine venner Lurvik og Smartnes har kommet over følgende algoritme, med beskrivelsen «How to sort a list in linear time»:

LINEAR-SORT(A, n)

- 1 $t_0 = \text{time}()$
- 2 MERGE-SORT($A, 1, n$)
- 3 SLEEP($1\,000\,000\,n - (\text{time}() - t_0)$)

De er begge enige om at det er litt «juks» å gjøre ting på den måten algoritmen gjør, men de har en litt filosofisk diskusjon om hva det er rimelig å si om kjøretiden. Lurvik mener man fint kan si at algoritmen har lineær kjøretid, mens Smartnes mener det er helt urimelig.

Hva mener du? Finn argumenter både for og imot. Diskuter og forklar.

Her er *time* en funksjon som gir nåværende tidspunkt, mens SLEEP er en prosedyre som venter et visst tidsintervall. Akkurat hva tidsenheten er er litt uklart.

- 10 Du har oppgitt en rettet graf $G = (V, E)$ og to noder $s, t \in V$, og skal avgjøre om det finnes en sti fra s til t i G . Problemet er at du har svært begrenset minne, som bare har plass til $O(\lg V)$ variable!

Én måte å bryte ned problemet må, kjent fra bl.a. korteste vei fra alle til alle, er å definere et sett med delproblemer ved hjelp av start- og sluttnode og *det maksimale antallet kanter vi får bruke, k* .

Beskriv en rekursiv dekomponering, og forklar hvordan denne kan brukes til å løse problemet, hvis du altså kun får bruke logaritmisk minne, men *så lang tid du vil*. Forklar svaret ditt.

Hint: Prøv å lage en rekursiv algoritme, der hvert rekursive kall har et konstant antall lokale variable, og der rekursjonsdybden er $O(\lg V)$.

Merk at variablene det er snakk om her, er typen som kan behandles i konstant tid i følge RAM-modellen (f.eks. boolske verdier eller heltall).

- 11 Du regner på næringsinnholdet i mat, og har representert informasjonen du har som en vektet, rettet, asyklisk graf $G = (V, E)$, der $(u, v) \in E$ betyr at måltidet eller matvaren u inneholder ingrediensen eller næringsstoffet v , og $w(u, v)$ angir mengden.

Du får oppgitt en mengde av en bestemt matvare, og skal skrive ut en oversikt over næringsinnholdet. Hvordan vil du gå frem? Forklar.

Hver node har sin egen enhet, så u kan f.eks. være «ml olivenolje», mens v er «g mettett fett», og $w(u, v) = 0,133$, som betyr at 15 ml olivenolje inneholder $15 \times 0,133 = 2$ g mettett fett. Du er ute etter det totale innholdet av *grunnleggende næringsstoffer*, representert av noder uten ut-kanter.

Merk at det kan være snakk om flere nivåer, der A inneholder B , som inneholder C , ... etc.

- 12 Innen fagfeltet *rettferdig fordeling* (*fair allocation*) finnes det et bestemt problem, der man vil fordele nodene i en graf på et sett med mottakere, så rettferdig som mulig, slik at nodene hver mottaker får utgjør en *sammenhengende graf*. Det kan f.eks. være snakk om å fordele kontor på ulike grupper, der kantene i grafen angir hvilke kontor som er nabokontor, og hver gruppe vil ha et sett med kontor som ligger samlet.

Du skal løse et spesialtilfelle av dette, der alle kontorene ligger på samme side i en korridor, så grafen utgjør en *sti*.

For enkelhets skyld, antar vi at alle gruppene er enige om hvor fine kontorene er, så hvert kontor får en *verdi* i form av et positivt tall. Vi ønsker å finne en *maksimin*-fordeling, som er rettferdig i den forstand at den gruppen som får den laveste totalverdien likevel får en så høy totalverdi som mulig.

Hvordan vil du løse dette problemet? Forklar.

Merk at vi her kun ser på *totalverdien* hver gruppe får, og ikke har noe krav til *antallet* kontor. Vi kan f.eks. anta at verdien henger sammen med størrelsen, dvs. antall personer det er plass til, så færre og bedre kontor er like ønskelig som flere og dårligere.