

TDT4120 Algoritmer og datastrukturer

Eksamen, 15. desember 2025, 15:00–19:00

Faglig kontakt Magnus Lie Hetland

Hjelpemiddelkode E

Oppgaver

- 5 % 1 Hva er kjøretiden til prosedyren for å bygge en maks-haug, BUILD-MAX-HEAP?
- 5 % 2 Din venn Lurvik har klart å vise at *to* av følgende beskrivelser av kjøretiden til en algoritme er korrekte, *men han husker ikke hvilke to*:
1. $T(n) = O(n^3)$
 2. $T(n) = \Omega(n^3)$
 3. $T(n) = \Theta(n^3)$
- Sjefen hans har bedt ham velge ut én av beskrivelsene, og nå har Lurvik bedt deg om hjelp. Hvilken av de tre velger du (nummer 1, 2 eller 3)? Forklar kort hvorfor dette er et godt valg.
- 5 % 3 Hva er det som gjør at COUNTING-SORT har lavere asymptotisk kjøretid enn f.eks. MERGE-SORT? Forklar kort.
- 5 % 4 Lurvik og Smartnes har en heftig diskusjon. Smartnes mener man kan ha en algoritme med ulik asymptotisk kjøretid i beste og verste tilfelle, der den gjennomsnittlige kjøretiden likevel er lik én av de to. Lurvik mener dette er umulig, siden gjennomsnittet y av x og z alltid vil ligge imellom dem, altså $x < y < z$.
- Bruk et eksempel fra pensum for å vise at Smartnes har rett. Forklar kort hvorfor det er mulig.
- Smartnes mener altså at *average-case* kan være lik enten *best-case* eller *worst-case*, selv om *best-case* og *worst-case* er forskjellige.
- 5 % 5 Dette er den rekursive formuleringen av lengden til den lengste felles delsekvensen (*longest common subsequence*, LCS) for prefiksene X_i og Y_j av X og Y .

$$c[i, j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ \text{[redacted]} & \text{if } i, j > 0 \text{ and } x_i = y_j, \\ \max\{c[i, j-1], c[i-1, j]\} & \text{if } i, j > 0 \text{ and } x_i \neq y_j. \end{cases}$$

Hva er det som mangler? Forklar kort.

- 5 % **6** Felles for følgende algoritmer er at hver node v har et bestemt felt av typen $v.x$ (men med et annet navn), som representerer en del av løsningen:

- DFS-VISIT (altså dybde-først-søk fra én node)
- BFS
- DAG-SHORTEST-PATHS
- DIJKSTRA
- BELLMAN-FORD
- PRIM

Hvilket felt er det snakk om (dvs., hva skal stå i stedet for « x » i « $v.x$ »)? Hva representerer det? Forklar kort.

Det er altså snakk om det samme feltet i alle algoritmene.

- 5 % **7** I skog-implementasjonen av disjunkte mengder (*disjoint-set forests*), som brukt bl.a. i KRUSKAL, hva skjer med foreldrepekerne når man bruker FIND-SET?

- 5 % **8** Dette er definisjonen av restkapasiteten (*residual capacity*) $c_f(u, v)$ i et flytnett (*flow network*) med flyt f :

$$c_f(u, v) = \begin{cases} \text{[redacted]} & \text{if } (u, v) \in E, \\ \text{[redacted]} & \text{if } (v, u) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Hva er det som mangler? Forklar kort.

- 5 % **9** Du studerer et problem P , og vil sammenligne det med et annet, kjent problem Q ved å finne en effektiv reduksjon. Hvilken vei vil du redusere for å vise hva? Forklar kort.

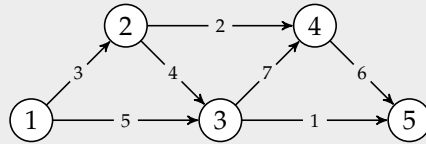
- 5 % **10** Hvordan kan antall sterke komponenter (*strongly connected components*) i en rettet graf endres hvis man legger til en ny kant? Forklar kort.

- 5 % **11** Hvorfor gir ikke DIJKSTRA nødvendigvis riktig svar dersom grafen vi bruker den på inneholder kanter med negativ vekt? Forklar kort.

Her må forklaringen være mer presis enn f.eks. at «valgene algoritmen gjør blir gale». Hvorfor blir de gale? Bruk gjerne et eksempel, om du har behov for det.

- 5 % **12** Hvor mye øker m om du utfører følgende prosedyre?

Figur 1



```
1 for i = 1 to n
2     m = m + 1
3     for j = 1 to n
4         m = m + 1
5         for k = 1 to n
6             m = m + 1
```

Oppgi svaret i Θ -notasjon, som funksjon av n . Forklar kort.

5% **13** Hvor mye øker m om du utfører følgende prosedyre?

```
1 k = 1
2 for i = 1 to n
3     m = m + k
4     k = k + k
```

Oppgi svaret i Θ -notasjon, som funksjon av n . Forklar kort.

5% **14** Løs følgende rekurrens eksakt:

$$T(n) = T(n - 1) + 2n - 1$$

$$T(0) = 0$$

Oppgi svaret uten asymptotisk notasjon. Forklar kort.

5% **15** Hva blir $T(n)$, om du løser følgende rekursive ligningssett?

$$T(n) = 2S(n/2) + (n \lg n)^2$$

$$S(n) = 8T(n/2) + n^2$$

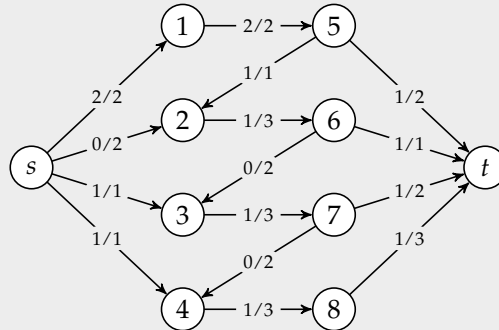
Oppgi svaret i Θ -notasjon. Forklar kort.

Bruk vanlige antagelser for algoritmiske rekurrenser.

5% **16** Utfør den første av de $|V| - 1$ iterasjonene til BELLMAN-FORD på grafen i figur 1, under følgende betingelser:

1. Bruk node 1 som startnode.
2. Anta at 8 er brukt i stedet for ∞ under initialiseringen.
3. Der du kan velge mellom kanter, velg den med lavest vekt.

Figur 2



Hva blir $v.d$ for hver node $v = 1, \dots, 5$ etterpå? Forklar kort.

Oppgi verdien for hver node, i rekkefølge.

5% **17** Figur 2 viser et flytnett (*flow network*) med flyt. Utfør én iterasjon av EDMONDS-KARP på flytnettet for å finne en forøkende sti (*augmenting path*). Oppgi nodene i den resulterende stien, i rekkefølge. Forklar kort.

5% **18** Nøklene (*keys*) i et søketre er vanligvis fra en ordnet mengde (f.eks. tall eller tekststrenger). Din venn Gløgsund har laget et søketre der hver node i stedet inneholder et *rektangel*. Når hun skal bygge treet, tar hun inn et sett S med ikke-overlappende rektangler, som legges i løvnodene. For hver indre node konstruerer hun et nytt rektangel: det minste som inneholder alle barnas rektangler.

Når hun skal søke i treet, tar hun inn et punkt, og sjekker rekursivt nedover i treet hvilke rektangler som inneholder punktet, og returnerer det av disse som ligger i en løvnode.

Gjør følgende antagelser:

- Rektanglene fordeles tilfeldig på løvnodene.
- Treet er perfekt balansert.

Hva blir kjøretiden for et søk, som funksjon av $n = |S|$? Gi både en øvre og nedre grense, dvs., bruk enten Θ -notasjon eller både O - og Ω -notasjon.

5% **19** Du utvikler et kodebibliotek for tekstprosessering, og skal implementere funksjonen `split(s, x)`, som deler strengen s ved alle forekomster av tegnet x . For eksempel vil

```
split("abc;de;fghi;jk", ";")
returnere
["abc", "de", "fghi", "jk"].
```

Du kan finne indeksene til x i s med en løkke. For en gitt slik indeks, kan du dele strengen i to mindre strenger, som er kopier av hver sin «halvdel».

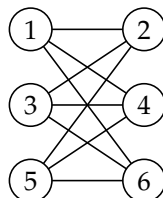
Om strengen du deler inneholder n tegn, må du kopiere $n - 1$ tegn. Konstruer og beskriv en algoritme som finner det minste totale antallet slike kopieringer som trengs. Hva blir kjøretiden? Forklar.

- 5% **20** For to grafer G og H er en *grafhomomorfi* $f : G \rightarrow H$ en funksjon f fra $G.V$ til $H.V$, som bevarer naboskap. For alle $u, v \in G.V$:

$$(u, v) \in G.E \implies (f(u), f(v)) \in H.E$$

Det vil si, hvis (u, v) er en kant i G , så er $(f(u), f(v))$ en kant i H . Å avgjøre om det eksisterer en homomorfi, gitt G og H , er NP-komplett generelt, men kan i visse tilfeller gjøres i polynomisk tid.

Anta at vi bestemmer at H alltid skal være følgende graf, kjent som $K_{3,3}$:



Input er nå bare G , og vi skal avgjøre om det finnes en homomorfi $f : G \rightarrow K_{3,3}$.

Enten vis at denne begrensede versjonen av problemet fortsatt er NP-komplett, eller vis hvordan problemet kan løses (dvs., avgjøres) i polynomisk tid. Kan du trekke noen generelle konklusjoner fra svaret ditt? Forklar.