

TDT4120 Algoritmer og datastrukturer

Eksamen, 7. august 2020, 09:00–13:00

Faglig kontakt Magnus Lie Hetland
Hjelpemiddelkode A

Oppgaver

- 1 Du har en tabell (*array*) med n heltall og skal finne de k største tallene. Noen løsninger vil ha lavere asymptotisk kjøretid mens andre vil være enklere å implementere, og ha lavere overhead på grunn av enklere datastrukturer, for eksempel. Diskuter mulige løsninger med ulike kjøretider, for eksempel $\Theta(kn)$, $\Theta(n \lg n)$, $\Theta(n \lg k)$ og $\Theta(n)$. Si litt om hvordan de fungerer, og hvilke fordeler og ulemper de har. Hvilke av løsningene dine vil fungere om du vil unngå ekstra minnebruk, og utføre operasjonene *in-place*, ved å bytte om på posisjonene til elementene? Hvilken av disse *in-place*-løsningene ville du ha brukt i en faktisk implementasjon?

Forklar og utdyp. Knytt til relevant teori, gjerne i ulike deler av pensum.

- 2 En artikkel fra femtitallet beskriver to problemer med tilhørende algoritmer: Å koble sammen alle nodene i en graf med minst mulig totalvekt, og å koble sammen to gitte noder med minst mulig totalvekt. Diskuter forholdet mellom problemene, og hvordan de løses. Hvilke likheter og forskjeller ser du? Er det forskjeller i hvilke grafer de fungerer på? Argumenter kort for svaret ditt.

Forklar og utdyp. Knytt til relevant teori, gjerne i ulike deler av pensum.

- 3 Når en algoritme utføres, gjentas gjerne et «trinn» mange ganger, til vi har funnet resultatet for en gitt instans. Hvordan relaterer dette seg til dekomponering av instansen i delinstanser (*subproblems*)? Hvordan arter dette seg forskjellig for ulike designmetoder? Hvilken rolle spiller matematisk induksjon oppi det hele? Hvilket slektskap har dekomponering til reduksjoner og hardhetsbevis, og hvordan stemmer forklaringen din overens med dette?

Forklar og utdyp. Knytt til relevant teori, gjerne i ulike deler av pensum.

- 4 Din venn Lurvik mener han har funnet på en ny algoritme for å finne korteste vei i vektete, rettede grafer, der vektene er positive heltall. Ideen hans er å transformere kanter (u, v) med vekt $w(u, v) = k > 1$ til stier $\langle u, x_1, x_3, \dots, x_{k-1}, v \rangle$ med lengde k , og så bruke BFS til å finne korteste vei. Hvilke fordeler og ulemper har denne metoden? Diskuter slektskap med algoritmer i pensum. Kunne du ha gjort noe lignende for å finne maksimal flyt med heltallskapasiteter, hvis du hadde en algoritme som kunne finne maksimal flyt? Hva slags algoritme måtte du i så fall ha hatt for å ta BFS sin plass?

Forklar og utdyp. Knytt til relevant teori, gjerne i ulike deler av pensum.

- 5 I grunnleggende kompleksitetsteori bruker vi gjerne NP som et slags «univers» for problemene vi ser på. Diskuter fordeler og ulemper med dette. Gi noen argumenter for at det er en stor klasse. Gi også eksempler på typer problemer som *ikke* er i NP, og som likevel har stor praktisk betydning. Gitt hva vi gjerne bruker klassen NP til, hvorfor er ikke dette så problematisk?

Forklar og utdyp. Knytt til relevant teori, gjerne i ulike deler av pensum.