

DM507 Algoritmer og datastrukturer

Forår 2010

Projekt, del I

Institut for matematik og datalogi
Syddansk Universitet

Februar 22, 2010

Dette projekt udleveres i tre dele. Hver del har sin deadline, således at afleveringerne, og dermed arbejdet, strækkes over hele semesteret. Projektet skal besvares individuelt. Deadline for del I er fredag den 19. marts.

Mål

Det samlede projekt beskæftiger sig med inversioner, og med sorteringsmetoder hvis køretid afhænger af antallet af inversioner i input.

Hovedmålet for del I af projektet er at lave en algoritme, som effektivt kan tælle antallet af inversioner i en følge.

Inversioner

En *inversion* i en følge $x_1, x_2, x_3, \dots, x_n$ af tal er et par x_i, x_j i følgen som indbyrdes ikke står i sorteret orden. I dette projekt defineres sorteret som stigende (mere præcist, ikke-faldende) orden, og en inversion er derfor et par x_i, x_j (med $i < j$) hvor $x_i > x_j$. *Antallet af inversioner i en følge* er antallet af par som udgør en inversion. Da der er $\binom{n}{2} = n(n-1)/2$ mulige par, er dette det maksimale antal inversioner i en følge med n elementer.

Eksempler: en stigende følge har 0 inversioner, en faldende følge har det maksimale antal inversioner $n(n-1)/2$, og denne følge

5, 2, 7, 1, 4

har 6 inversioner (parrene 5,2; 5,1; 5,4; 2,1; 7,1 og 7,4).

Algoritmer til at tælle inversioner i en følge

En oplagt algoritme til at finde antallet af inversioner i en følge bruger en dobbelt **for**-løkke, og tager derfor $\Theta(n^2)$ tid. I del I af projektet skal vi udvikle en algoritme, som tager $\Theta(n \log n)$ tid. Algoritmen er en udvidelse af MergeSort algoritmen, og er baseret på følgende observationer:

1. Hvis en følge $x_1, x_2, x_3, \dots, x_n$ deles i to

$$A = x_1, x_2, x_3, \dots, x_k \quad B = x_{k+1}, x_{k+2}, \dots, x_n$$

da er det samlede antal inversioner lig $X + Y + Z$, hvor:

X er antallet af inversioner i A

Y er antallet af inversioner i B

Z er antallet af inversioner x_i, x_j med $i \leq k$ og $j \geq k + 1$

2. Z ændres ikke selv om A og B hver især sorteres.

Opgaver

1. Argumenter for korrektheden af observationerne ovenfor.
2. Beskriv en udvidelse af metoden MERGE (side 31 i Cormen et al.) som udover at flette to sorterede følger A og B til én sorteret følge også beregner tallet Z . Argumenter for korrektheden af din metode. [Hint: Se på en inversion x_i, x_j som hørende til det højre endepunkt x_j . Brug også den del af løkke-invarianten for MERGE (side 32 i Cormen et al.) som siger, at hvad der allerede er kopieret fra L og R , er mindre end eller lig med alt, der ikke er kopieret endnu.]
3. Beskriv en algoritme baseret på MergeSort som finder antallet af inversioner i en følge i $\Theta(n \log n)$ tid. [Hint: brug de to observationer ovenfor.] Argumenter for korrekthed og køretid din algoritme.
4. Implementer den simple $\Theta(n^2)$ algoritme baseret på en dobbelt **for**-løkke til at finde antallet af af inversioner i en følge. Implementationen skal ligge i en fil der hedder **SimpleInv.java**. Den skal tage sit input fra en tekstfil, der indeholder en følge af heltal, hver adskilt af whitespace (brug f.eks. klassen `java.util.Scanner` og metoden `nextInt()` herfra til at parse input). Den skal tage inputfilens navn fra kommandolinien, således at programmet efter kompilering kan kaldes således: `java SimpleInv inputfilnavn`. Den skal skrive output (antal inversioner i input) på skærmen. Der skal udskrives et simpelt tal og intet andet (da de afleverede programmer vil blive afprøvet med automatiserede tests).
5. Implementer MergeSort efter pseudo-koden i afsnit 2.3 i Cormen et al. Som ∞ kan bruges værdien `Integer.MAX_VALUE` fra klassen `Integer`. Det må her og i resten af projektet antages, at denne værdi ikke forekommer i input. Implementationen skal ligge i en fil der hedder **MergeSort.java**. Den skal tage sit input fra en tekstfil, der indeholder en følge af heltal, hver adskilt af whitespace. Den skal tage inputfilens navn fra kommandolinien, og skal skrive output på skærmen (som de sorterede tal, hver adskilt af whitespace, og intet andet).
6. Implementer din $\Theta(n \log n)$ algoritme baseret på MergeSort til at finde antallet af af inversioner i en følge. Implementationen skal ligge i en fil der hedder **FastInv.java**. Den skal tage sit input fra en tekstfil, der indeholder en følge af heltal, hver adskilt af whitespace. Den skal tage inputfilens navn fra kommandolinien, og skal skrive output (antal inversioner i input, og intet andet) på skærmen.

Formalia

Lav en rapport, som indeholder dine svar på opgaverne 1–6 ovenfor. Koden fra opgaverne 4–6 skal være passende kommenteret, skal inkluderes i rapporten som bilag, og eventuelle ikke-trivielle aspekter af implementeringen skal diskuteres i rapportens hoveddel. Der skal afleveres rapporten i pdf-format, samt de tre Java-programmer som separate filer (dvs. udover deres inklusion på tryk i rapporten).

Materialet afleveres med **aflever** kommandoen i en shell på Imadas Linux-system: Lav et directory som indeholder ovenstående fire filer, flyt ned i dette, og udfør kommandoen **aflever DM507**. Dette vil kopiere indholdet af dette directory til et sted i systemet, som underviseren kan tilgå. Man kan bruge kommandoen flere gange, senere anvendelser overskriver materialet fra tidligere anvendelser.

Aflever materialet senest:

Fredag den 19. marts, 2010, kl. 23:59.