

# DM02 Eksamen – Obligatorisk Opgave

## Evolutionstræer

### Første del: Editeringsafstand

## 1 Problemet

I denne del-opgave skal I bruge dynamisk programmering til at beregne forskellen eller *editeringsafstanden* mellem to strenge. Editeringsafstanden mellem to strenge  $x$  og  $y$  er det mindste antal editeringsoperationer, man kan klare sig med, hvis man vil ændre  $x$  til  $y$  eller omvendt. En editeringsoperation består i at indsætte et tegn, slette et tegn eller ændre et tegn til et andet tegn. De tre operationer kaldes *indsæt*, *slet* og *erstat*.

Beregning af editeringsafstand bruges i mange sammenhænge: stavetekorrektion, database-søgninger og evolutionære, strukturelle og funktionelle sammenligninger af biologiske strenge som DNA-sekvenser og proteinstrengene.

### 1.1 Et lille eksempel

Lad os se på de to ord “datalogi” og “algoritme”. Det kræver 8 editeringsoperationer at ændre “datalogi” til “algoritme”. Til gengæld kan det gøres på flere forskellige måder med netop 8 operationer. Man kan f.eks. slette d, a og t, indsætte g, erstatte g med r og indsætte t, m og e, man kan slette d, t og a, indsætte g, erstatte g med r og indsætte t, m og e, eller man kan slette d, erstatte t med l og a med g, slette l, erstatte g med r og indsætte t, m og e. Disse tre muligheder er illustreret nedenfor (I betyder “indsæt”, S betyder “slet”, og E betyder “erstat”), men der er også andre muligheder.

SSS I E III	S SS I E III	S EES E III
datal ogi	datal ogi	datalogi
algoritme	a lgoritme	alg oritme

## 2 Programmet

I skal løse problemet v.h.a. dynamisk programmering, og programmet skal have køretid  $O(mn)$  og pladsforbrug  $O(\min\{m, n\})$ , hvor  $m$  og  $n$  er længderne af de to input-strengene. I anden del-opgave, som udleveres i november, skal I bygge videre på dette program.

**Input:** De to strenge, der skal sammenlignes lægges i en fil med to linier; en streng per linie. Input-filen svarende til eksemplet i afsnit 1.1 ser altså sådan ud:

```
datalogi
algoritme
```

**Output** fra programmet skal være et enkelt tal, som angiver editeringsafstanden mellem de to input-strengene. For eksemplet i afsnit 1.1 vil output være tallet 8.

Hovedklassen skal hedde EditDistance (stort “E” og “D”). Man skal kunne afvikle programmet på følgende måde:

```
java EditDistance <strengfil>
```

hvor filen <strengfil> indeholder de to strenge som beskrevet ovenfor.

## 3 Hjælp

### 3.1 Indlæsning af filer

Indlæsning af filer kan klares med

```
FileReader fr = new FileReader(fileName);
BufferedReader inFile = new BufferedReader(fr);
```

hvorefter man kan læse en linie ad gangen med

```
s = inFile.readLine();
```

### 3.2 Test

Hvis I gerne vil have en ide om, hvad vi senere vil sige til det output, I producerer, kan I skrive DM02check. I skal, før I afgiver kommandoen, placere jer i det katalog, som alle jeres oversatte Java-programmer ligger i. Så vil jeres program blive kørt på samtlige testfiler i /home/IMADA/courses/dm02/Tests. Hvis der findes fejl, bliver dette skrevet på standard output, ellers skrives der intet. Hvis der ikke findes fejl, betyder det ikke nødvendigvis, at jeres program fungerer helt korrekt, da testen ikke er udtømmende. Programmet DM02check kan afbrydes med Ctrl-c.

## 4 Krav til rapport og program

Først og fremmest skal alle krav beskrevet i denne opgaveformulering naturligvis tilfredsstilles.

Programmet skal være velstruktureret og kommenteret i passende omfang.

Rapporten skal indeholde en udskrift af hele programmet. Denne udskrift skal være identisk med det elektronisk afleverede program.

Rapporten skal indeholde en beskrivelse af de væsentligste valg, der er truffet i forbindelse med implementationen, samt begrundelser herfor.

Man skal argumentere for, at den samlede køretid af programmet er  $O(mn)$ , og at pladsforbruget er  $O(\min\{m, n\})$ .

Desuden skal det forklares, hvordan programmet er afprøvet. Det er ikke tilstrækkeligt blot at have kørt DM02check, da denne test ikke er udtømmende.

Eventuelle mangler i program eller rapport skal beskrives.

Endelig skal rapporten underskrives.

## 5 Aflevering

Der skal afleveres en rapport på sekretariatet, og et program skal afleveres elektronisk, begge dele inden mandag d. 27. oktober 2003 kl. 10:00.

Den elektroniske aflevering foregår på følgende måde: Opret et katalog (directory), som indeholder alle jeres JAVA-filer til opgaven og intet andet. Stil jer i dette katalog, og afgiv kommandoen `aflever DM02`. Dette skal ske fra maskinen `tosca`.

D.v.s. gør følgende.

```
ssh toska (afgiv password, hvis der bliver bedt om det)
cd <opgave-katalog>
aflever DM02
```

hvor `opgave-katalog` indeholder alle jeres JAVA-filer til opgaven og intet andet.

Bemærk, at I (inden afleveringsfristen) kan aflevere flere gange, hvis I fortryder en aflevering. Kun den sidste aflevering tæller (de andre slettes).

Rapporten skal afleveres på sekretariatet på Institut for Matematik og Datalogi. Denne opgaveformulering er vedhæftet en forside, som skal anvendes ved afleveringen. Husk for jeres egen skyld at få en kvittering (også vedhæftet) på, at I har afleveret.

Husk, at det er et krav, at det elektronisk afleverede program er præcis det samme program, der afleveres en udskrift af i rapporten.

## 6 Evaluering

For at kunne gå til eksamen i DM02 til januar skal man have bestået den obligatoriske opgave. Har man tidligere lavet en obligatorisk DM02-opgave og fået den godkendt, er det tilstrækkeligt.

For denne første del af den obligatoriske opgave er der kun to mulige bedømmelser: "godkendt" eller "ikke godkendt". Selvom første del bliver godkendt, kan der være ting, som skal rettes inden aflevering af anden del-opgave for, at den samlede opgave kan godkendes. Hvis bedømmelsen er "ikke godkendt", kan man ikke aflevere anden del og kan altså ikke deltage i eksamen.

## 7 Yderligere formalia

Den obligatoriske opgave er et individuelt eksamensprojekt. En overtrædelse af dette er eksamenssnyd og vil blive behandlet som sådan. Man har pligt til selv at beskytte sine noter og filer mod læsning af andre. Dette kan f.eks. gøres med `chmod 700 <opgavedirectory>`. Begge parter involveret i en eventuel plagiering kan blive holdt ansvarlige.

Programmet skal kunne køre på IMADAs maskiner. Man må gerne lave det hjemme, men det er helt på eget ansvar. Tekniske problemer derhjemme er ingen undskyldning. Man er også helt selv ansvarlig for, at ens filer kan flyttes uden problemer.

Der SKAL afleveres til tiden. Undtagelser herfra kræver særlige omstændigheder, som vil kunne holde i studienævnet. Vi har pligt til at behandle jer ens.

God fornøjelse!



DM02 eksamen, Efteråret 2003  
Obligatorisk Opgave — Første Del

**Skriv tydeligt** (maskinskrift eller blokbogstaver)

Navn: .....

Fødselsdato: .....

Brugernavn (login): .....

<b>Instruktør</b>	Fiona	Frederik	Martin	Steffen
<b>Sæt kryds</b>				

Besvarelsen omfatter ..... nummererede sider.



**Kvittering for aflevering af  
obligatorisk opgave, første del, i DM02, efteråret 2003**

**Udfyldes inden afleveringen**

Navn: .....

Fødselsdato: .....

Brugernavn (login): .....

**Udfyldes af sekretariatet**

Modtaget den ..... kl. .... af .....

(dato)                      (klokken)                      (initialer)