Algoritmer og Datastrukturer 2

Aflevering 3

Kristian Gausel[1], Lasse Alm[2], and Steffan Sølvsten[3]

[1]201509079@post.au.dk
[2]201507435@post.au.dk
[3]201505832@post.au.dk

22. december 2016

1 Opgave 5

For de tre strenge x, y og z værende

$$x = x_1 x_2 ... x_n$$
$$y = y_1 y_2 ... y_m$$
$$z = z_1 z_2 ... z_{n+m}$$

så kaldes z et *flet* af x og y, hvis x og y kan findes som to disjunkte delsekvenser i z, og hvis kun disse to strenge tilsammen udgør hele z.

For $0 \leq i \leq n$ og $0 \leq j \leq m$ er $F[i, j]$ en boolsk værdi, der angiver om strengen $z_1 z_2 ... z_{i+j}$ er et flet af strengene $x_1 x_2 ... x_i$ og $y_1 y_2 ... y_j$. For $i = 0$, $j = 0$ eller $i + j = 0$, så defineres den respektive streng som den tomme streng.

Der oplyses følgende rekursionsformel for at bestemme $F[i, j]$

$$F[i, j] = \begin{cases} X_{ij} \vee Y_{ij} & , \ i, j \geq 1 \\ X_{ij} & , \ i \geq 1, j = 0 \\ Y_{ij} & , \ i = 0, j \geq 1 \\ \text{Sand} & , \ i, j = 0 \end{cases} \tag{1}$$

Udsagnene X_{ij} og Y_{ij} er

$$X_{ij} = (z_{i+j} = x_i \wedge F[i-1, j]) \tag{2}$$
$$Y_{ij} = (z_{i+j} = y_i \wedge F[i, j-1]) \tag{3}$$

1.1 A - Tabel for x = uro, y = gled og z = gulerod

Som eksempel sættes $x = uro$ og $y = gled$ og $z = gulerod$. Værdier for F er vist i tabel 1.

	y	0	1	2	3	4
x			G	L	E	D
0		**S**	**S**	F	F	F
1	U	F	**S**	**S**	**S**	F
2	R	F	F	F	**S**	F
3	O	F	F	F	**S**	**S**

Tabel 1: Tabellen for $F[i, j]$ opskrevet. Indgange med S = Sand og F = Falsk

1.2 B - Fletning bestemt ved brug af dynamisk programmering

En algoritme, der er baseret på dynamisk programmering, og som har en tidskompleksitet på $O(nm)$, skal bestemmes og opskrives som pseudokode.

Ved at bruge rekursionsformlen for F, så kan resultatet nemt beregnes, dog vil man bemærke, at der vil være flere rekursive kald til beregning af $F[i, j]$ for samme værdier af i og j. Dette kan løses ved brug af memoization, hvor resultater af allerede beregnede indgange gemmes i en matrice tilsvarende til tabel 1 i afsnit 1.1. Ved alene at tilføje i rekursionen en test for om en værdi allerede er gemt i denne matrice, så er det muligt at reducere antallet af beregninger af én indgang til kun det først rekursive kald. Med $x.length = n$ og $y.length = m$, så er matricen $n \cdot m$ stor, hvormed der laves $O(n \cdot m)$ rekursive kald, som alle tager konstant tid at udføre. Hermed er tidskompleksiteten ved nedsat til $O(nm)$.

Algoritmen er opdelt i to funktioner, $Flet(x, y, z)$ i kode 1, hvor matricen dannes, som derefter udfyldes af funktionen $F(i, j)$ i kode 2.

Kode 1: Kode for algoritmen Flet(x, y, z)

```
Flet(x, y, z)
    //The matrix Q, corresponding to the same purpose as table 1
    Q = Matrix of size (x.length, y.length) with all entries NIL

    return F(x.length, y.length)
```

Den rekursive formel er implementeret i funktionen $F(i,j)$, der tilgår og udfylder matricen Q fra $Flet(x,y,z$ ved at tilgå x, y og z fra samme scope.

Kode 2: Kode for hjælpe algoritmen F(i, j)

```
F(i, j)
    if Q[i,j] != NIL
        return Q[i,j]

    else if i == 0 and j == 0
        Q[i,j] = true

    else if i == 0 and j ≥ 1
        Q[i,j] = F(i, j-1) and (z[i+j] == y[j])

    else if i ≥ 1 and j == 0
        Q[i,j] = F(i-1, j) and (z[i+j] == x[i])

    else if i ≥ 1 and j ≥ 1
        Q[i,j] = (F(i-1, j) and z[i+j] == x[i]) or
                  (F(i, j-1) z[i+j] == y[j])

    return Q[i,j]
```

1.3 C - Udskrivning af indekserne for løsning

En udvidelse til algoritmen fra afsnit 1.2 skal findes, der også returnerer indekserne for delsekvenserne af z, som er lig x.

Såfremt et flet af x og y er fundet for z, så vil der være mindst en sti af *sande* værdier gennem matricen Q. Hver sti gennem Q er en løsning til fletningen af x og y for z. Følges en specifik sti, så kan alle indeks for delsekvenserne x af z kunne findes findes som indgange (i,j), hvor $(i-1,j)$ også er sand. Indekset for denne indgang i z er $i+j$, hvormed alle indeks for en af løsningerne for x i z kan udskrives ved brug af algoritmen i kode 3.

Kode 3: Kode for at udprinte indeks for delsekvenser af z, som er lig x

```
PrintFlet(Q, i, j)
    if i == 0 and j == 0
        return

    else if Q[i-1, j] == true
        PrintFlet(Q, i-1, j)
        print i + j

    else //if Q[i,j-1] == true
        PrintFlet(Q, i, j-1)
```

Idet der for hvert rekursive kald enten fjernes en kolonne eller en række i matricen af størrelse $n \times m$, så vil tidskompleksiteten for udprintningen være $O(n+m)$.