Exam subject outline - Nicklas Jacobsen qmr656

De fire generale step i konstruktionen af en dynamisk programerings algoritme er:

1. Karakterisere strukturen i en optimal løsning
2. Rekursivt finde de optimale løsninger til delproblemerne
3. Beregne den optimale værdi for det overordnede problem
4. Konstruerer en optimal løsning til fra den beregnede information

De tre krav til dynamistisk programerings algoritmer:

1. For at man kan bruge dynamisk programering, skal de optimale løsninger til sub-problemerne være en delmængde af den optimale løsning til det oprindelige problem.
2. To sub-problemer til det samme problem skal være uafhænige af hinanden.
3. Sub-problemerne skal overlappe hinanden, dvs. mængden af forskellige sub-problemer skal være relativ lille (modsat f. eks. divide and conquer hvor man generer nye sub-problemer ved hvert step.)

LCS eksempel

LCS er den længste fælles del-sekvens af to stringe. Hvis vi har to stringe $X = \{x_1, x_2..., x_m\}$ og $Y = \{y_1, y_2..., y_n\}$ og deres LCS $Z = \{z_1, z_2...z_k\}$, så gælder det at:

Optimal sub-struktur

1. Hvis $x_m = y_n$, så $z_k = x_m = y_n$ og $Z_k - 1$ er en LCS af X_{m-1} og Y_{n-1}
2. Hvis $x_m \neq y_n$, så $z_k \neq x_m$ som indikerer at Z er en LCS af X_{m-1} og Y
3. Hvis $x_m \neq y_n$, så $z_k \neq y_n$ som indikerer at Z er en LCS af Y_{n-1} og X

Bevis 1 *Første del:*
Hvis $x_m = y_n$ og $x_m \neq z_k$ så kunne vi tilføje $x_m = y_n$ til Z og få en LCS af X og Y på længden $k+1$, hvilket ville være i modstrid med at Z er en LCS af X og Y. Det må derfor være at hvis $x_m = y_n$ så $z_k = x_m = y_n$.
Anden del: Vi ønsker at bevise at hvis $x_m = y_n$ så er Z_{k-1} en LCS af X_{m-1} og Y_{n-1}. Med formål for modstrid antager vi, at der eksisterer en fælles sekvens W for X_{m-1} og Y_{n-1} som har en længde større end $k-1$. Hvis vi tilføjer $x_m = y_n$ til W resulterer det i en fælles sekvens for X og Y, med en længde større end k, hvilket er i modstrid med at Z er en LCS.

Bevis 2 og vice versa for 3 Hvis $x_m \neq z_k$ så er Z en LCS af X_{m-1} og Y, for hvis der eksisterer en fælles sekvens W for X_{m-1} og Y med længde større end k, så vil W også være en LCS af X og Y, hvilket ville være i modstrid med at Z er en LCS af X og Y.

Overstående viser at LCS problemet har en optimal-substruktur.

Overlappende sub-problemer

Ved at kigge på de tre punkter under "Optimal sub-struktur", kan man se at for at finde LCS af X og Y skal vi håndtere en ud af to situationer:
1: Hvis $x_m = y_n$ så skal vi finde LCS for X_{m-1} og Y_{n-1}, og ved at tilføje $x_m = y_n$ til denne får vi LCS af X og Y.
2: I tilfældet af at $x_m \neq y_n$, skal vi løse to problemer. Vi skal finde LCS af X_{m-1} og Y, og vi skal finde LCS af X og Y_{n-1}, og retunerer den LCS der er længst.
Det tydeligt at se at vi kommer til at løse de samme sub-problemer flere gange.

Resultat

	j	0	1	2	3
i			**A**	**D**	**C**
0		0	0	0	0
1	**A**	0	$\nwarrow$ 1	$\leftarrow$1	$\leftarrow$1
2	C	0	$\uparrow$1	$\leftarrow$1	$\nwarrow$2
3	**D**	0	$\uparrow$1	$\nwarrow$2	$\leftarrow$2
4	A	0	$\nwarrow$1	$\uparrow$2	$\leftarrow$2
5	**C**	0	$\uparrow$1	$\uparrow$2	$\nwarrow$3

Tabellen ovenfor viser resultatet af en dynamisk programmerings algortime til at beregne LCS af to stringe i og j. For at aflæse tabellen, starter man nede i nederste højre hjørne og følger pilene. Hver gang man møder $\nwarrow$ tilføjer man det pågældende bogstav til sin LCS.