Institut for Matematik og Datalogi
Syddansk Universitet, Odense

1. april 2022
RF

Opgaver Uge 14

SE4-DMAD

A: Løses i løbet af øvelsestimerne i uge 14

Opgaverne 1–6 er repetition af tidligere stof.

1. Eksamen juni 2016, opgave 1.
2. Eksamen juni 2016, opgave 2.
3. Eksamen juni 2016, opgave 3.
4. Eksamen juni 2016, opgave 7.
5. Eksamen juni 2016, opgave 8.
6. Eksamen juni 2016, opgave 9.
7. Cormen et al. øvelse 15.1-3 (side 370).
8. Cormen et al. øvelse 15.4-1 (side 396).
9. Cormen et al. øvelse 15.4-2 (side 396).
10. Eksamen juni 2010, opgave 4.
11. Eksamen juni 2013, opgave 7.

B: Løses hjemme inden øvelsestimerne i uge 16

1. Cormen et al. opgave 2.3 (side 41). Hint: for at overskue situationen, prøv algoritmen på konkrete instanser, og skriv summen i spørgsmål **c** ud.

2. (∗) Cormen et al. øvelse 4.2-7 (side 83). Produktet af to komplekse tal $a + bi$ og $c + di$ er $(ac - bd) + (ad + bc)i$, dvs. opgaven består i at beregne $ac - bd$ og $ad + bc$ ud fra a, b, c og d, men ved kun at bruge tre multiplikationer. Svaret ligner lidt Strassens metode (men er meget simplere).

3. Cormen et al. øvelse 15.4-5 (side 397). Hint: lav en (1-dimensional) tabel over $l(i)$ for $i = 1$ til n, hvor $l(i)$ angiver længden af en længste monotont stigende delsekvens *som ender* ved det i'te tal. Dvs. løs dette lidt modificerede problem med dynamisk programmering. Brug til sidst tabellen med løsningerne for dette problem til at løse det oprindelige problem.

   [At bruge ovenstående metode er meningen med opgaven. Man kan faktisk også finde længste monotont stigende delsekvens ved at se på inputsekvensen som en streng af tal, og derefter løse et LCS-problem mellem strengen selv og en udgave af den som er blevet sorteret, således at alle tallene kommer i stigende rækkefølge (udfordring: bevis at dette løser det samme problem). Derefter kan man bruge at vi jo allerede har en metode baseret på dynamisk programmering til at løse LCS. Er køretiden den samme eller er den forskellig for de to måder?]

4. Cormen et al. problem 15-2 (side 405). Hint: For at løse dette direkte med dynamisk programmering, lad delproblemer være givet ved delstrengen $x_i x_{i+1} \ldots x_{j-1} x_j$ for $i \leq j$, og se i analysen på begge ender (dvs. på x_i og x_j) af delproblemet, når man forsøger at relatere det til mindre delproblemer. Herudover ligner analysen lidt den for Longest Common Subsequence.

   [Ovenstående er meningen med opgaven. Man kan faktisk også løse dette palindrom-problem ved at løse LCS-problemet for strengen selv og dens omvendte (hård udfordring: bevis dette), hvilket vi jo allerede har en metode baseret på dynamisk programmering for. Er køretiden den samme eller er den forskellig for de to måder?]