

Eksamens Disposition: Ugeopgave 2

Dynamisk programmering (Dynamic programming)

Sarah Kirstine Willumsen

Overordnet formål

Dynamisk programmering er en algoritme eller teknik, der er velegnet til at optimere køretiden, når man skal løse problemer der kan deles op i mindre problemer. Dette gøres ved at gemme løsningen til et underproblem i en tabel, så man undgår at genberegne identiske underproblemer, da løsningen så bare kan hentes fra tabellen i konstant-tid.

Metoden

Vi følger 4 step når vi udvikler en dynamisk programmerings algoritme.

1. Karakteriser strukturen for en optimal løsning dvs. find den optimale understruktur.

Note: se s. 11 i tavle noter

2. Rekursivt definere værdien af en optimal løsning.

Note: s. 12 + 13 i tavle noter.

3. Beregn værdien af en optimal løsning f.eks. ved at bruge rekursion.

Note: s. 14-15 i tavle noter.

4. Konstrukturer den optimale løsning.

Note: s. 23 og frem i tavle noter.

Eksempel: rod cutting

Dynamisk programmering er velegnet at bruge til optimerings problemer.

Et eksempel er det der kaldes rod cutting problemmet: finde den optimal opdeling.

Vi vil finde maks samlede fortjeneste r_n dvs. finde den optimale under opdelling.

afprøve alle måder længden n kan opdeles på.

Der er 2^{n-1} måder at kombinere opdelingen på, fordi vi laver 2 deling (cut eller no-cut) $n-1$ gange. Så det vil give en algortime med en eksponential køretid $T(n) = 2^{n-1}$

Vis at få køretid $\theta(n^2)$ ved Dynamisk programmering.

To metoder til implemtering: top-down med memoization (Gennemgå første algoritme på s. 366) og bottom-up metoden (Gennemgå anden algoritme s. 366).

Fordele og ulemper s. 22 i tavle noter.

Bottom-up løser alle under problemer

Memoized løser kun de under problemmmmer som er nødvendige.

LCS: The longest commen subsequence problem s. 25 - 30 i tavle noter

Evt. perspektiver

Greedy algorithms s. 414