

Eksamensdisposition - Grådige algoritmer

Søren Mulvad, rbn601

3. april 2019

- **Paradigmet**
- **Håndkørsel af Huffman**
 - Eksempel: {a:45, d:16, b:13, c:12, e:9, f:5}
- **Bevis af korrekthed af Huffman**
 - Redegørelse for Lemma omkring summen af frekvenser
 - Greedy choice property
 - Optimal delstruktur

1 Grådige algoritmer

Paradigmet

- Grådige algoritmer minder en smule om dynamisk programmering, i og med vi har en række delproblemer der afhænger af løsningen på andre delproblemer.
- Forskellen er, at med en grådig algoritme laver vi altid den løsning som på nuværende tidspunkt ser bedst ud, uden at kigge på fremtidige delproblemer.
- Hermed laver vi en lokal optimal løsning, som vi håber vil lede til en global optimal løsning.

- **Egenskaber for problemerne:**

Greedy choice property:

Vil sige at vi kan samle en global optimal løsning ved at lave lokale optimale (grådige) valg. Altså, når vi beregner hvilket valg vi skal tage, kan vi tage det valg som er bedst for det nuværende problem uden at tage højde for løsninger til delproblemerne. Hermed får vi hele tiden et lignende, men mindre, delproblem.

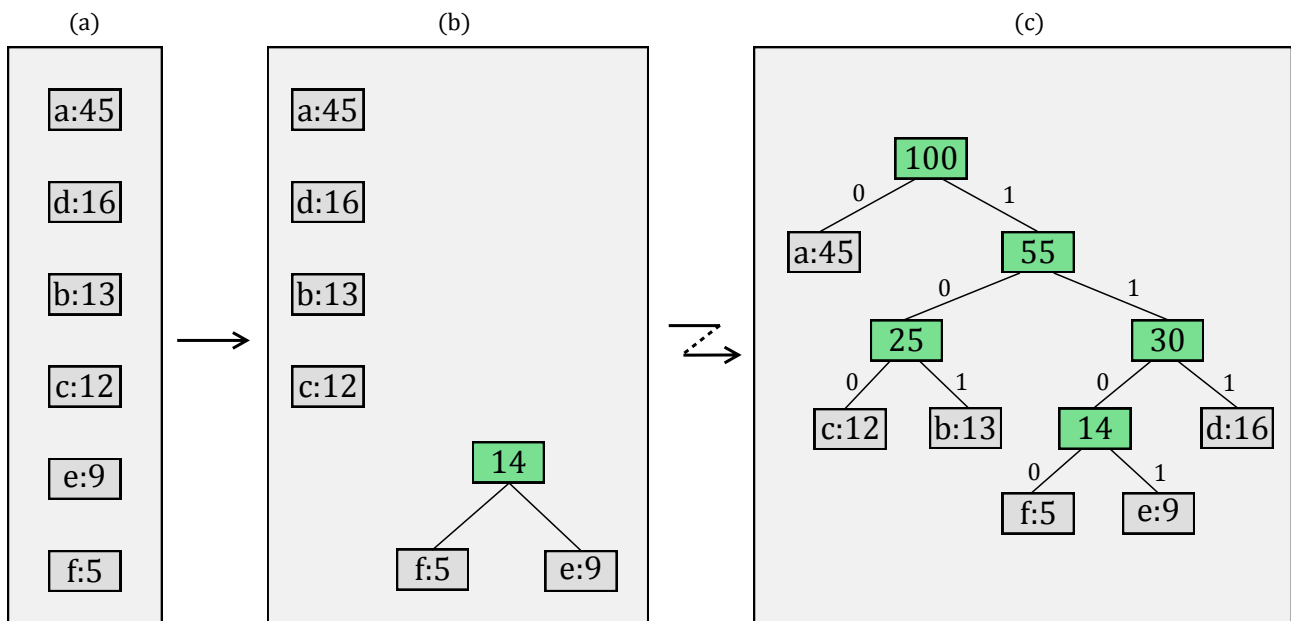
Optimal delstruktur:

Vise sige, at hvis der fandtes en optimal løsning der indeholder det grådige valg, så består den optimale løsning af det grådige valg + en løsning til det delproblem vi har tilbage når vi har lavet det grådige valg.

- Eksempler på grådige algoritmer er activity-selection, Dijkstras algoritme og Huffman-koder. Jeg vil her gå mere i dybden med Huffman-koder.

Håndkørsel af Huffman

- Løser problemet med at vi gerne vil sende noget information, men ønsker at gøre brug af, at der er en hvis redundans i informationen. Ved at kode de karakterer der optræder flest gange med den laveste kode får vi den mindst mulige filstørrelse. Men hvordan finder vi ud af, hvordan vi skal kode dem, så vi overholder dette samtidig med at slutkodningen ikke kan tolkes på flere måder?
- Vi indfører følgende notation:
 - T er et parse-træ for en præfikskode.
 - C er sættet af karakterer, f.eks. $C = \{a, b, c, d, e, f\}$. For hver karakter $c \in C$ har vi:
 - f_c for frekvensen af denne karakter.
 - $d_T(c)$ for dybden af bladet i T .
 - $B(T) = \sum_{c \in C} f_c \cdot d_T(c)$ er omkostningen af træet. Vi ønsker naturligvis at minimere $B(T)$.
- Algoritmen virker ved at tage de to elementer med lavest frekvens og sætte dem sammen så de bliver søskende i et deltræ med en forælder der har en frekvens som er summen af dens børn. Dette gør algoritmen $n - 1$ gange, så alle knuder til sidst er sat sammen til et og samme deltræ.



Figur 1: (a) Algoritmen idet alle karakterer $c \in C$ er sat ind i vores datastruktur, men inden vi har kørt nogle iterationer. (b) Algoritmen efter første iteration, hvor vi vælger f og e fordi de har lavest frekvens. De sættes sammen og der laves en forældreknode med frekvensen $f_f + f_e = 5 + 9 = 14$. (c) Algoritmen efter alle $n - 1$ iterationer af for-loopet er kørt og vi har vores færdige parse-træ T . Nu ville bitrepræsentationen for f.eks. $a = 0$ og $f = 1100$.

- Køretiden afhænger af hvor hurtigt vi kan finde elementet med lavest frekvens og tage det ud, samt hvor hurtigt vi kan indsætte nye elementer. Hvis vi f.eks. bruger Fibonacci Heaps kan vi både køre **Extract-Min** og **Insert** i $O(\lg n)$ tid. Vi ser derudover, at for hver iteration fjerner vi to elementer og tilføjer et, altså er der netto ét element mindre at håndtere hver gang, hvorved vi får $\Theta(n)$ iterationer. Således bliver den totale køretid:

$$O(n \lg n)$$

Bevis af korrekthed af Huffman

Vi har lige bestemt et parse-træ T ved at bruge Huffmans algoritme. Nu vil jeg gerne bevise, at $B(T)$ er minimal. Vi skal bruge et Lemma jeg ikke decideret vil bevise, men som giver god mening ud fra Figur 1.

Lemma: Hvis vi lader W være alle knuder på nær roden, så har vi for ethvert parse-træ T at:

$$B(T) = \sum_{c \in C} f_c \cdot d_T(c) = \sum_{v \in W} f_v$$

Dvs. vores frekvens-dybde sum er lig summen af alle frekvenserne af knuder der ligger i W . Se f.eks på (c) i Figur 1.

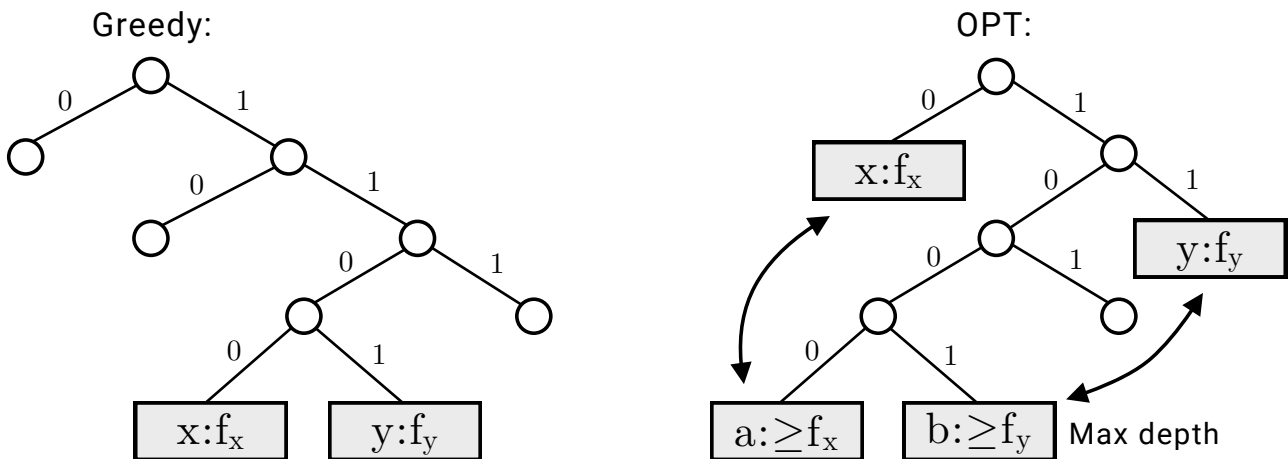
For nu at bevise, at **Huffman** giver en optimal præfikskode skal vi vise greedy choice property og optimal delstruktur.

Greedy choice property:

Vi skal altså vise, at det grådige valg er med i mindst en af de optimale løsninger. Dvs. når vi finder de to symboler med lavest frekvens, sætter dem sammen, og giver dem en fælles forælder, så findes der en optimal løsning som har de her to knuder med mindst frekvens som søskendeblade i den optimale løsning.

Lad os kalde de to karakterer med lavest frekvens for x og y .

Vi har greedy choice property fra det faktum, at et optimalt træ OPT kan blive transformeret om til et andet træ OPT' som har det grådige valg (at x og y er søskendeblade):



Figur 2: Bemærk der med a og b i ulighederne i figuren menes frekvensen af de to, altså f_a og f_b

På [Figur 2](#) har vi først outputtet fra **Huffman** (dvs. x og y er søskendeblade) og derefter et arbitrært optimalt træ OPT . Men OPT kan vi nu lave om til et nyt træ OPT' der også har det grådige valg. Det kan vi, fordi ulighederne på figuren gælder da vi netop har defineret x og y til at være de elementer med henholdsvis lavest og næstlavest frekvens.

Frekvensen af a er mindst lige så stor som frekvensen af x . Så ved at flytte a op kan vi få en kortere bitrepræsentation til noget der optræder lige så mange gange eller hyppigere end x . Her ses et formelt bevis, hvor $OPT = T$ er træet før ombytningen og $OPT' = T'$ er træet efter ombytningen af x og y :

$$\begin{aligned} B(T') &= B(T) - f_a \cdot d_T(a) - f_x \cdot d_T(x) \\ &\quad + f_a \cdot d_T(x) + f_x \cdot d_T(a) \end{aligned} \quad (1)$$

$$= B(T) + \underbrace{(f_a - f_x)}_{>0} \underbrace{(d_T(x) - d_T(a))}_{<0} \quad (2)$$

$$\leq B(T) \implies T' \text{ er optimal} \quad (3)$$

Eq. Eq. (1) Vi ved, at omkostningen af træet T' hvor x byttes med a svarer det til det gamle træ, og hvor vi så tager højde for de ændringer vi laver. Vi tager udgangspunkt i det optimale træ T i alle led.

Eq. Eq. (2) Vi sætter de to frekvenser sammen til et led samt de to dybder til et led. Det første led er ≥ 0 da det er vores antagelse jf. vores greedy choice property. Det andet led er ≤ 0 , da vi valgte a til at have maksimal dybde.

Eq. Eq. (3) Vi ser, at vi muligvis ganger et positivt med et negativt tal, og derfor må det led totalt blive ≤ 0 . Således får vi $B(T') \leq B(T)$, som medfører at T' er optimal.

Altså har vi nu vist, at det grådige valg hvor x og y er søskendeblade er med i mindst en af de optimale løsninger.

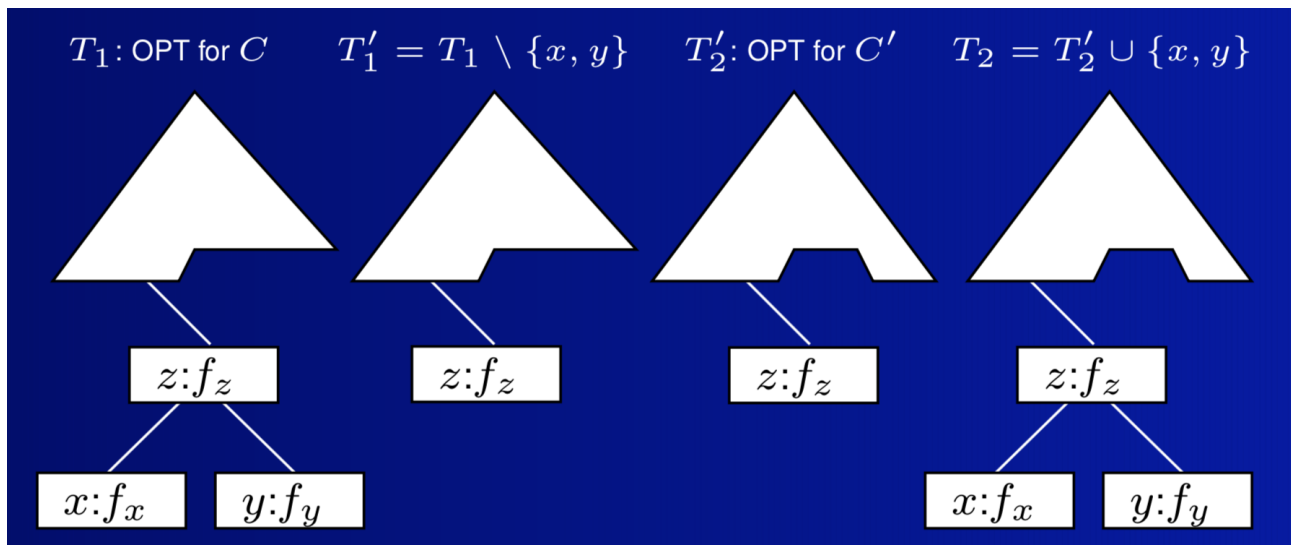
Optimal delstruktur:

Det vi nu vil vise er, at hvis der fandtes en optimal løsning der indeholder det grådige valg, så består den optimale løsning af det grådige valg + en løsning til det delproblem vi har tilbage når vi har lavet det grådige valg.

Det delproblem vi har tilbage er et nyt alfabet $C' = (C \setminus \{x, y\}) \cup \{z\}$, altså hvor vi har fjernet x og y fra det oprindelige problem og indsat en ny "karakter" z med $f_z = f_x + f_y$.

Nu skal vi vise, at hvis det grådige valg er i et optimalt træ T_1 for C , så er $T'_1 = T_1 \setminus \{x, y\}$ et optimalt træ for C' .

Derudover indfører vi T'_2 , som er en optimal løsning til vores problem med det reducerede alfabet C' , og $T_2 = T'_2 \cup \{x, y\}$ som er den optimale løsning med søskendebladene.



Vi ønsker at vise, at $B(T'_1) \leq B(T'_2)$, da T'_2 netop er defineret til at være en optimal løsning for delproblem C' . Ved at vise dette viser vi nemlig, at deløsningen T'_1 faktisk er en optimal løsning for C' .

Fra figuren ser vi:

$$B(T'_1) = B(T_1) - f_x - f_y \leq B(T_2) - f_x - f_y = B(T'_2)$$

Lighederne gælder, da vi jf. vores Lemma ved at $B(T)$ er lig summen af frekvenserne af alle knuder på nær roden.

Uligheden gælder, da vi definerede at T_1 er en optimal løsning til C .