Aflevering 3
Algoritmer og datastrukturer
Forår/Sommer 2011

Naja Mottelson (vsj465)
Søren Pilgård (vpb984)

12. maj 2011

Indhold

1 a

1.1 Algoritme

Vi ønsker at løse opgaven at give byttepenge (bestående af pennies, nickels, dimes og quarters) for n cents benyttende det mindst mulige antal mønter. En grådig algoritme til at gøre dette kan være som følger: Hvis $n = 0$ gives 0 mønter. Hvis $n > 0$ findes den største mønt med værdi c, $c <= n$, denne gives og derefter rekurserer vi ved at finde byttepenge for $n - c$ cents.

1.2 Bevis

For at bevise at denne algoritme giver den optimale løsning beviser vi indledningsvist at den grådige egenskab gælder. I dette tilfælde vil det betyde at en given optimal løsning altid vil indeholde en mønt med værdien c, hvor $c <= n$. For at gøre dette forestiller vi os en optimal løsning. Såfremt denne løsning er nødt til at indeholde en mønt med værdien c behøver vi ikke gøre mere. Ellers antager vi at løsningen ikke indeholder en mønt med værdien c. Vi har følgende fire grænsetilfælde for n:

$1 <= n < 5$: I dette tilfælde vil $c = 1$. Eftersom denne optimale løsning udelukkende må indeholde pennies, må den nødvendigvis indeholde en mønt med værdien c.

$5 <= n < 10$: Her vil $c = 5$. Denne optimale løsning vil ligeledes udelukkende indeholde pennies, eftersom vi antager at den ikke indeholder en nickel. Antallet af pennies vil dog nødvendigvis være større end 5, hvorved vi kan erstatte de 5 pennies med en nickel og få en løsning med 4 færre mønter.

$10 <= n < 25$: Her vil $c = 10$. Vi antager at denne optimale løsning ikke indeholder en dime. En en delmængde af de pennies og nickels den indeholder vil dog kunne summeres til 10, som så kan erstattes med en dime og give en løsning med færre mønter (det specifikke antal færre mønter ligger imellem 1 og 9).

$25 <= n$: Her vil $c = 25$. Ligesom ovenfor antager vi at denne optimale løsning ikke indeholder en quarter. Den værdi (som er skarpt større end n) som indeholder det færrest mulige antal mønter vil i dette tilfælde være 3 dimes. Disse vil kunne erstattes af to dimes og en nickel og så give en optimal løsning med færre mønter.

Således har vi vist at en optimal løsning altid indeholder det grådige valg (c). Køretid for denne algoritme er $\Theta(k)$ hvor $k =$ antallet af mønter i en optimal løsning. Vi ved at $k <= n$, så algoritmen er $O(n)$.

2 b

Vi har møntenhederne $c^0, c^1, c^2, \ldots c^k$ for heltallene $c > 1$ og $k >= 1$. En grådig algoritme til at give byttepenge for n cents kunne i dette tilfælde virke ved at finde den møntenhed c^j der passer på at $j = max(0 <= i <= k)$ hvor $c^i <= n$. Den returnerer en mønt c^j og rekurserer over underproblemet at give byttepenge

for $n - n^j$.
For at kunne føre beviset beviser vi først følgende lemma: Lad a_i være antallet af mønter brugt i en given optimal løsning. Da gælder at $a_i < c$.
Bevis: Hvis $a_i >= c$ for et i hvor $0 <= i < k$ så kan vi forbedre løsningen ved at bruge én mere mønt af enhed $c^i + 1$ og c færre mønter møntenhed c^i. Vi vil i dette tilfælde få en løsning med $c - 1$ færre mønter.
Vi betragter nu en ikke-grådig løsning, som ikke benytter nogen mønter af møntenhed c^j eller højere. Lad den ikke-grådige løsning bruge a_i mønter af møntenhed c^i for $i = 0, 1, \ldots j - 1$.
Vi kan, pr. ovenstående lemma, konkludere at en optimal løsning ikke kan indeholde mere end $(c - 1)$ af hver møntenhed. Således vil den maksimale værdi vi kan konstruere vha. mønter af møntenhed $<= c^j - 1$ kunne beskrives som $(c-1)c^0 + c-1)c^1 + c-1)c^2 + \ldots c-1)c^k$. Vi ser at denne sumfølge kan skrives op på formen for geometriske serier (A.5 i CLRS):

$$(c-1)\sum_{i=0}^{j} -1c^i = \frac{c^j - 1}{c - 1}(c-1)$$

Vi ser endvidere at dette kan reduceres til

$$(c-1)\sum_{i=0}^{j} -1c^i = c^j - 1$$

Eftersom vi ved at $c^j <= n$, og således at $c^j - 1 < n$ kan vi ud fra ovenstående konkludere at man ikke vil være i stand til at give byttepenge for n cents udelukkende vha. mønter af møntenhederne $c^0, c^1, c^2, \ldots c^j - 1$. Dette betyder at den ikke-grådige løsning ikke er optimal.
Eftersom enhver algoritme som ikke inkluderer det grådige valg (c^j) må være suboptimal, kan vi konkludere at den grådige løsning altid vil være optimal.

3 c

Fjerner vi 'nickel'-enheden fra mængden af enheder i denne opgave får vi en mængde af møntenheder hvorpå den grådige strategi ikke giver en optimal løsning. Eksempel: $n = 30$. Her vil en grådig algoritme give én quarter og fem pennies, hvor den optimale løsning ville have været 3 dimes.

4 d

Lad $num[j]$ = min. antal mønter vi skal bruge for at give byttepenge for j cents og de forskellige møntenheder være $d_0, d_1, d_2 \ldots d_k$.

Hvis vi vidste at en optimal løsning indeholdt en møntenhed di, ville vi have $num[j] = 1 + num[j - di]$.

Lad $den[j]$ være en møntenhed benyttet i en optimal løsning for at give byttepenge for j cents.

Vores algoritme skal returnere num og den, da disse tabeller indeholder al den information vi behøver for at udregne den optimale løsning.

Udkast til algoritme:

```
CHANGE(n, d, k)
```

```
for j <- 1 to n:
    num[j] <- infinity
        for i <- 1 to k:
            if j >= $d_i$  and 1 + num[j - $d_i$] < num[j]
                then num[j] <- 1 + num[j - $d_i$]
                den[j] <- $d_i$
return num, den
```