Øvingsforelesning 8

TDT4120 - Algoritmer og datastrukturer

Øving 7

Grådige algoritmer - Egenskaper

Oppgave 2: Hvilke egenskaper trenger vi for at et problem kan løses av en grådig algoritme?

- Kun én optimal løsning på problemet
- Grådighetsegenskapen (*the greedy-choice property*)
- Overlappende delinstanser (*subproblems*)
- Optimal delstruktur

Grådige algoritmer - Egenskaper

Oppgave 2: Hvilke egenskaper trenger vi for at et problem kan løses av en grådig algoritme?

- ~~Kun én optimal løsning på problemet~~
- Grådighetsegenskapen (*the greedy-choice property*)
- Overlappende delinstanser (*subproblems*)
- Optimal delstruktur

Grådige algoritmer - Egenskaper

Oppgave 2: Hvilke egenskaper trenger vi for at et problem kan løses av en grådig algoritme?

- ~~Kun én optimal løsning på problemet~~
- Grådighetsegenskapen (*the greedy-choice property*)
- ~~Overlappende delinstanser (*subproblems*)~~
- Optimal delstruktur

Grådige algoritmer - Egenskaper

Oppgave 2: Hvilke egenskaper trenger vi for at et problem kan løses av en grådig algoritme?

- ~~Kun én optimal løsning på problemet~~
- Grådighetsegenskapen (*the greedy-choice property*)
- ~~Overlappende delinstanser (*subproblems*)~~
- Optimal delstruktur

Oppgave 3: Hva vil det si at et problem har grådighetsegenskapen?

Grådige algoritmer - Egenskaper

Oppgave 2: Hvilke egenskaper trenger vi for at et problem kan løses av en grådig algoritme?

- ~~Kun én optimal løsning på problemet~~
- Grådighetsegenskapen (*the greedy-choice property*)
- ~~Overlappende delinstanser (*subproblems*)~~
- Optimal delstruktur

Oppgave 3: Hva vil det si at et problem har grådighetsegenskapen?

At vi kan sette sammen en globalt optimal løsning på problemet ved å gjøre lokalt optimale valg.

Grådige algoritmer - Løsbare problemer

Oppgave 4: Hvilke av disse problemene kan løses med en grådig algoritme?

- Stavkutting
- Aktivitets-utvelgelse
- Det binære ryggsekkproblemet
- Det kontinuerlige ryggsekkproblemet
- Lengste felles delsekvens

Grådige algoritmer - Løsbare problemer

Oppgave 4: Hvilke av disse problemene kan løses med en grådig algoritme?

- ~~Stavkutting~~
- Aktivitets-utvelgelse
- Det binære ryggsekkproblemet
- Det kontinuerlige ryggsekkproblemet
- Lengste felles delsekvens

Grådige algoritmer - Løsbare problemer

Oppgave 4: Hvilke av disse problemene kan løses med en grådig algoritme?

- ~~Stavkutting~~
- Aktivitets-utvelgelse
- ~~Det binære ryggsekkproblemet~~
- Det kontinuerlige ryggsekkproblemet
- Lengste felles delsekvens

Grådige algoritmer - Løsbare problemer

Oppgave 4: Hvilke av disse problemene kan løses med en grådig algoritme?

- ~~Stavkutting~~
- Aktivitets-utvelgelse
- ~~Det binære ryggsekkproblemet~~
- Det kontinuerlige ryggsekkproblemet
- ~~Lengste felles delsekvens~~

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	3	4	5	6	7	8
Startid	2	2	5	10	8	1	4	5
Sluttid	3	4	8	12	16	5	7	11

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av Recursive-Activity-Selector?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

```
Recursive-Activity-Selector(s, f, k, n)
m = k + 1
while m ⩽ n and s[m] < f[k]
    m = m + 1
if m ⩽ n
    return {a_m} ∪ Recursive-Activity-Selector(s, f, m, n)
else return ∅
```

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)

$m = k + 1$
while $m \leqslant n$ and $s[m] < f[k]$
    $m = m + 1$
if $m \leqslant n$
    return $\{a_m\} \cup$ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
else return $\emptyset$

k = 0, n = 8, aktiviteter = {}

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av Recursive-Activity-Selector?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

Recursive-Activity-Selector(s, f, k, n)

$m = k + 1$
while $m \leqslant n$ and $s[m] < f[k]$
    $m = m + 1$
if $m \leqslant n$
    return $\{a_m\} \cup$ Recursive-Activity-Selector(s, f, m, n)
else return $\emptyset$

k = 1, n = 8, aktiviteter = $\{a_1\}$

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)

```
m = k + 1
while m ⩽ n and s[m] < f[k]
      m = m + 1
if m ⩽ n
      return {a_m} ∪ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
else return ∅
```

k $=$ 4, n $=$ 8, aktiviteter $= \{a_1, a_7\}$

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)

```
m = k + 1
while m ⩽ n and s[m] < f[k]
      m = m + 1
if m ⩽ n
      return {a_m} ∪ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
else return ∅
```

k $=$ 8, n $=$ 8, aktiviteter $= \{a_1, a_7, a_4\}$

Aktivitets-utvelgelse

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

Oppgave 6: Hvor mange aktiviteter kan man maksimalt velge?

Oppgave 5: For aktivitetene under, hvilken aktivitet ville blitt valgt som aktivitet nummer to av RECURSIVE-ACTIVITY-SELECTOR?

Aktivitet #	1	2	6	7	3	8	4	5
Startid	2	2	1	4	5	5	10	8
Sluttid	3	4	5	7	8	11	12	16

Oppgave 6: Hvor mange aktiviteter kan man maksimalt velge?

Maksimalt 3.

Alternativ aktivitet-sutvelgelse

Oppgave 7: Hver aktivitet har en verdi. Vi ønsker å makismere verdien på aktivitetene vi velger, heller enn antall. Finner RECURSIVE-ACTIVITY-SELECTOR alltid en optimal løsning på dette problemet?

Oppgave 8: Bevis at svaret i oppgave 7 stemmer.

Alternativ aktivitet-sutvelgelse

Oppgave 7: Hver aktivitet har en verdi. Vi ønsker å makismere verdien på aktivitetene vi velger, heller enn antall. Finner RECURSIVE-ACTIVITY-SELECTOR alltid en optimal løsning på dette problemet?

Oppgave 8: Bevis at svaret i oppgave 7 stemmer.

Aktivitet #	Starttid	Sluttid	Verdi
1	1	3	3
2	2	5	4

Alternativ aktivitet-sutvelgelse

Oppgave 7: Hver aktivitet har en verdi. Vi ønsker å makismere verdien på aktivitetene vi velger, heller enn antall. Finner RECURSIVE-ACTIVITY-SELECTOR alltid en optimal løsning på dette problemet?

Oppgave 8: Bevis at svaret i oppgave 7 stemmer.

Aktivitet #	Starttid	Sluttid	Verdi
1	1	3	3
2	2	5	4

RECURSIVE-ACTIVITY-SELECTOR velger aktivitet 1.

Alternativ aktivitets-utvelgelse

Oppgave 9: Hver aktivitet har en frist og en varighet. Ønsker å fullføre flest mulig aktiviteter før fristen. Gir følgende algoritme en optimal løsning?

1. Lag et tomt tidsskjema.
2. Sorter aktivitetene etter synkende frist.
3. Iterer gjennom aktivitetene. Finnes det minst et tidspunkt hvor en aktivitet kan utføres før fristen, plasser den på det siste slike tidpunktet.

Oppgave 10: Bevis at svaret i oppgave 9 stemmer.

Alternativ aktivitets-utvelgelse

Oppgave 9: Hver aktivitet har en frist og en varighet. Ønsker å fullføre flest mulig aktiviteter før fristen. Gir følgende algoritme en optimal løsning?

1. Lag et tomt tidsskjema.
2. Sorter aktivitetene etter synkende frist.
3. Iterer gjennom aktivitetene. Finnes det minst et tidspunkt hvor en aktivitet kan utføres før fristen, plasser den på det siste slike tidpunktet.

Oppgave 10: Bevis at svaret i oppgave 9 stemmer.

Aktivitet #	1	2	3
Frist	11	6	8
Varighet	11	3	5

Alternativ aktivitets-utvelgelse

Oppgave 9: Hver aktivitet har en frist og en varighet. Ønsker å fullføre flest mulig aktiviteter før fristen. Gir følgende algoritme en optimal løsning?

1. Lag et tomt tidsskjema.
2. Sorter aktivitetene etter synkende frist.
3. Iterer gjennom aktivitetene. Finnes det minst et tidspunkt hvor en aktivitet kan utføres før fristen, plasser den på det siste slike tidpunktet.

Oppgave 10: Bevis at svaret i oppgave 9 stemmer.

Aktivitet #	1	2	3
Frist	11	6	8
Varighet	11	3	5

1	2	3	4	5	6	7	8
2	2	2	3	3	3	3	3

Huffmankoding - Programmering

Oppgave 11: Implementer en funksjon som gitt en tekststreng og en oppslagstabell med huffmankodinger returnerer huffmankodingen av tekststrengen.

Huffmankoding - Programmering

Oppgave 11: Implementer en funksjon som gitt en tekststreng og en oppslagstabell med huffmankodinger returnerer huffmankodingen av tekststrengen.

```
ENCODE(data, encoding)
encoded = ""
for i = 1 to data.length
    encoded = encoded + encoding[data[i]]
return encoded
```

Huffmankoding - Programmering

Oppgave 12: Implementer en funksjon som gitt et huffmantre konstruerer en oppslagstabell for huffmankodingene til symbolene i treet.

Huffmankoding - Programmering

Oppgave 12: Implementer en funksjon som gitt et huffmantre konstruerer en oppslagstabell for huffmankodingene til symbolene i treet.

```
ENCODING(node, prefix, encoding)
if node.character
    encoding[node.character] = prefix
else
    ENCODING(node.left-child, prefix + "0", encoding)
    ENCODING(node.right-child, prefix + "1", encoding)
```

Huffmankoding

Oppgave 13: Gitt frekvensene for bokstavene under, hva blir huffmankoden til C?

Bokstav	A	B	C	D
Frekvens	2	2	5	10

Huffmankoding

Oppgave 13: Gitt frekvensene for bokstavene under, hva blir huffmankoden til C?

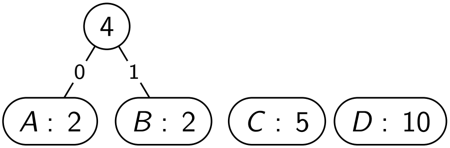
Bokstav	A	B	C	D
Frekvens	2	2	5	10

$A : 2$ $B : 2$ $C : 5$ $D : 10$

Huffmankoding

Oppgave 13: Gitt frekvensene for bokstavene under, hva blir huffmankoden til C?

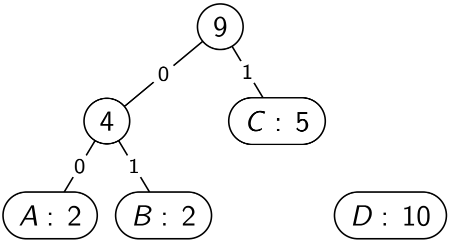
Bokstav	A	B	C	D
Frekvens	2	2	5	10



Huffmankoding

Oppgave 13: Gitt frekvensene for bokstavene under, hva blir huffmankoden til C?

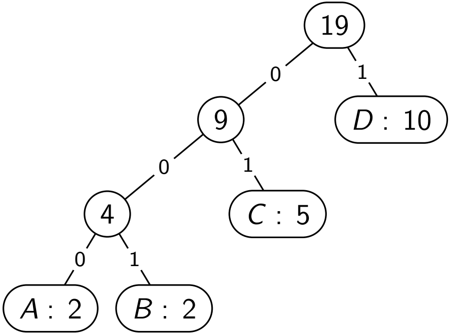
Bokstav	A	B	C	D
Frekvens	2	2	5	10



Huffmankoding

Oppgave 13: Gitt frekvensene for bokstavene under, hva blir huffmankoden til C?

Bokstav	A	B	C	D
Frekvens	2	2	5	10



Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116

Huffmankoding

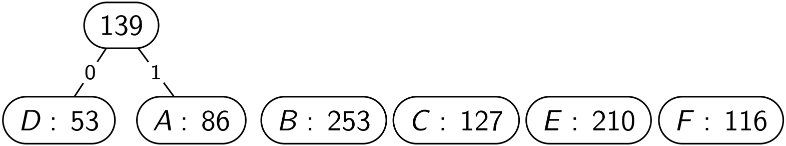
Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116

$A:86$ $B:253$ $C:127$ $D:53$ $E:210$ $F:116$

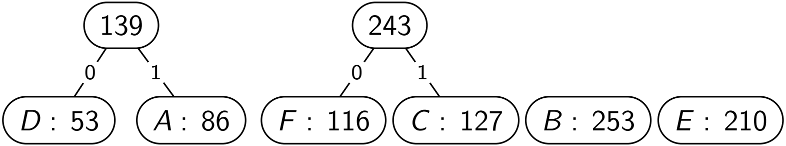
Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116



Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

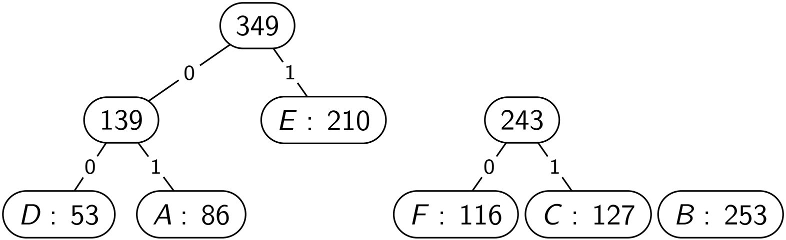
Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116



Huffmankoding

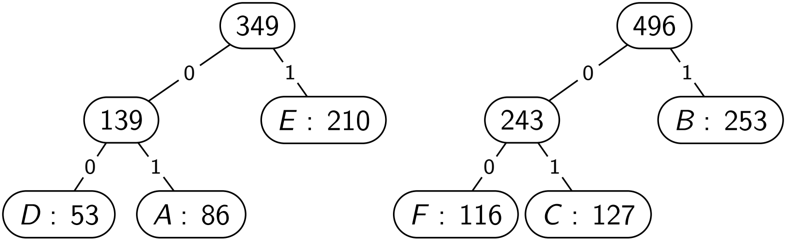
Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116



Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

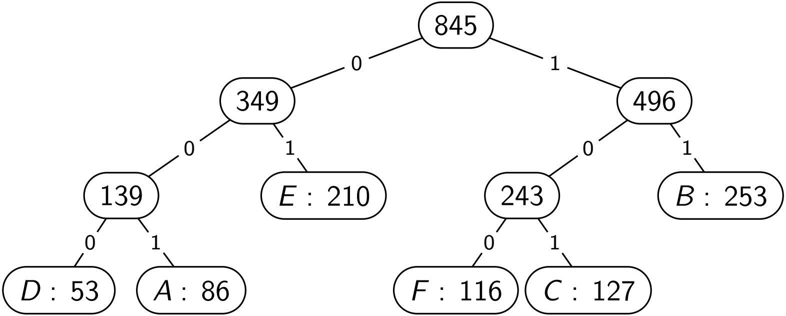
Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116



Huffmankoding

Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

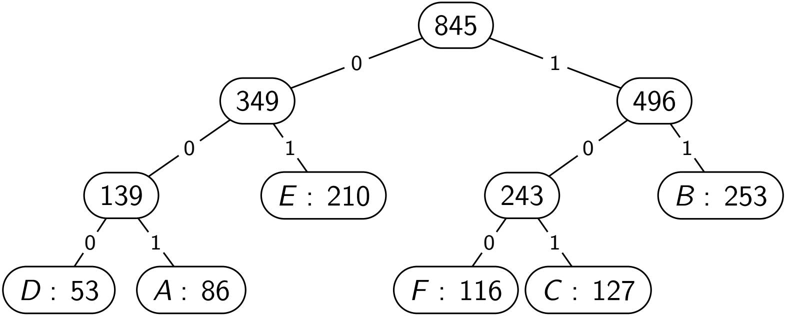
Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116



Huffmankoding

Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116
Huffmankode	001	11	101	000	01	100



Huffmankoding

Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116
Huffmankode	001	11	101	000	01	100

$$B(T) = \sum_{c \in C} c.\mathit{freq} \cdot d_T(c)$$

Huffmankoding

Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116
Huffmankode	001	11	101	000	01	100

$$
\begin{aligned}
B(T) &= \sum_{c \in C} c.\mathit{freq} \cdot d_T(c) \\
&= 86 \cdot 3 + 253 \cdot 2 + 127 \cdot 3 + 53 \cdot 3 + 210 \cdot 2 + 116 \cdot 3
\end{aligned}
$$

Huffmankoding

Oppgave 14: Gitt frekvensene for bokstavene under, hva blir lengden på den huffmankodede filen?

Bokstav	A	B	C	D	E	F
Frekvens	86	253	127	53	210	116
Huffmankode	001	11	101	000	01	100

$$
\begin{aligned}
B(T) &= \sum_{c \in C} c.\mathit{freq} \cdot d_T(c) \\
&= 86 \cdot 3 + 253 \cdot 2 + 127 \cdot 3 + 53 \cdot 3 + 210 \cdot 2 + 116 \cdot 3 \\
&= 2072
\end{aligned}
$$

Huffman

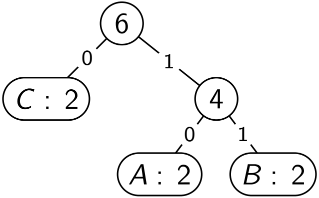
Oppgave 15: Hva stemmer om HUFFMAN?

- Et tegn vil aldri få en lengre kode enn et annet tegn med strengt lavere frekvens.
- Man er garantert at resultatet er en optimal prefikskode.
- To tegn som har samme frekvens vil alltid få like lang kode.
- Hvis et tegn har strengt større frekvens enn de andre tegnene til sammen, vil dette tegnet få en kode med lengde 1.
- Et huffmantre vil alltid ha høyde $\Theta(\lg n)$ dersom det er n unike tegn i teksten.

Huffman

Oppgave 15: Hva stemmer om HUFFMAN?

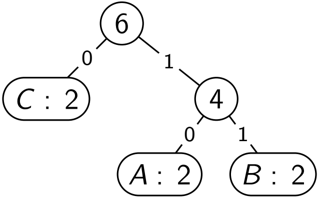
- Et tegn vil aldri få en lengre kode enn et annet tegn med strengt lavere frekvens.
- Man er garantert at resultatet er en optimal prefikskode.
- To tegn som har samme frekvens vil alltid få like lang kode.
- Hvis et tegn har strengt større frekvens enn de andre tegnene til sammen, vil dette tegnet få en kode med lengde 1.
- Et huffmantre vil alltid ha høyde $\Theta(\lg n)$ dersom det er n unike tegn i teksten.



Huffman

Oppgave 15: Hva stemmer om HUFFMAN?

- Et tegn vil aldri få en lengre kode enn et annet tegn med strengt lavere frekvens.
- Man er garantert at resultatet er en optimal prefikskode.
- ~~To tegn som har samme frekvens vil alltid få like lang kode.~~
- Hvis et tegn har strengt større frekvens enn de andre tegnene til sammen, vil dette tegnet få en kode med lengde 1.
- Et huffmantre vil alltid ha høyde $\Theta(\lg n)$ dersom det er n unike tegn i teksten.



Oppgave 15: Hva stemmer om HUFFMAN?

- Et tegn vil aldri få en lengre kode enn et annet tegn med strengt lavere frekvens.
- Man er garantert at resultatet er en optimal prefikskode.
- ~~To tegn som har samme frekvens vil alltid få like lang kode.~~
- Hvis et tegn har strengt større frekvens enn de andre tegnene til sammen, vil dette tegnet få en kode med lengde 1.
- Et huffmantre vil alltid ha høyde $\Theta(\lg n)$ dersom det er n unike tegn i teksten.

Symbol	s_1	s_2	s_3	$\dots$	s_n
Frekvens	1	2	4	$\dots$	2^{s_n}

Oppgave 15: Hva stemmer om HUFFMAN?

- Et tegn vil aldri få en lengre kode enn et annet tegn med strengt lavere frekvens.
- Man er garantert at resultatet er en optimal prefikskode.
- ~~To tegn som har samme frekvens vil alltid få like lang kode.~~
- Hvis et tegn har strengt større frekvens enn de andre tegnene til sammen, vil dette tegnet få en kode med lengde 1.
- ~~Et huffmantre vil alltid ha høyde $\Theta(\lg n)$ dersom det er n unike tegn i teksten.~~

Symbol	s_1	s_2	s_3	$\dots$	s_n
Frekvens	1	2	4	$\dots$	2^{s_n}

Beslutningstrær - Programmering

Oppgave 16: Gitt et sett med klassifikasjoner og deres sannsynlighet. Konstruer et beslutningstre som minimaliserer hvor mange spørsmål som må stilles i gjennomsnitt.

Oppgave 16: Gitt et sett med klassifikasjoner og deres sannsynlighet. Konstruer et beslutningstre som minimaliserer hvor mange spørsmål som må stilles i gjennomsnitt.

```
BUILD-DECISION-TABLE(decisions)
1  T = HUFFMAN(decisions)
2  return ENCODING(T)
```

Mynter

Oppgave 17: Vi skal representere et vilkårlig beløp ved bruk av færrest mulig mynter. Myntene kan ha følgende verdier $\{1, 2, 5, 10, 20, 50, 100\}$. Vil det fungere å grådig plukke den mynten med høyest verdi som ikke overstiger gjenstående beløp?

Mynter

Oppgave 17: Vi skal representere et vilkårlig beløp ved bruk av færrest mulig mynter. Myntene kan ha følgende verdier $\{1, 2, 5, 10, 20, 50, 100\}$. Vil det fungere å grådig plukke den mynten med høyest verdi som ikke overstiger gjenstående beløp?

En optimal løsning kan kun ha én ener, to toere, en femmer, en tier, to mynter med verdi 20 og en med verdi 50.

Mynter

Oppgave 17: Vi skal representere et vilkårlig beløp ved bruk av færrest mulig mynter. Myntene kan ha følgende verdier $\{1, 2, 5, 10, 20, 50, 100\}$. Vil det fungere å grådig plukke den mynten med høyest verdi som ikke overstiger gjenstående beløp?

En optimal løsning kan kun ha én ener, to toere, en femmer, en tier, to mynter med verdi 20 og en med verdi 50.

Algoritmen vil alltid produsere en optimal løsning.

Mynter

Oppgave 17: Vi skal representere et vilkårlig beløp ved bruk av færrest mulig mynter. Myntene kan ha følgende verdier $\{1, 2, 5, 10, 20, 50, 100\}$. Vil det fungere å grådig plukke den mynten med høyest verdi som ikke overstiger gjenstående beløp?

Oppgave 18: La myntene nå være $\{1, 15, 30, 45, 50, 60, 100\}$.

Mynter

Oppgave 17: Vi skal representere et vilkårlig beløp ved bruk av færrest mulig mynter. Myntene kan ha følgende verdier $\{1, 2, 5, 10, 20, 50, 100\}$. Vil det fungere å grådig plukke den mynten med høyest verdi som ikke overstiger gjenstående beløp?

Oppgave 18: La myntene nå være $\{1, 15, 30, 45, 50, 60, 100\}$.

For beløpet 65 trenger man to mynter: $50 + 15$.
Algoritmen finner seks mynter: $60 + 1 + 1 + 1 + 1 + 1$.

Asyklisk graf

Oppgave 19: Du ønsker å gjøre en syklisk urettet graf asyklisk, men vil ha at de gjenværende kantene har størst mulig samlet vekt. Kan dette løses grådig?

Asyklisk graf

Oppgave 19: Du ønsker å gjøre en syklisk urettet graf asyklisk, men vil ha at de gjenværende kantene har størst mulig samlet vekt. Kan dette løses grådig?

1. Sortert kantene i stigende rekkefølge basert på vekt.
2. Gå igjennom kantene. Sjekk om hver kant er en del av en sykel, i så fall fjern kanten.

Brønnbygging

Oppgave 20: Er sett med landsbyer ligger langs en vei. Skal bygge brønner, men færrest mulig. Avstanden fra en landsby til nærmeste brønn kan maksimalt være r. Finn en algoritme for plassering av brønnene med kjøretid $o(n^2)$.

Brønnbygging

Oppgave 20: Er sett med landsbyer ligger langs en vei. Skal bygge brønner, men færrest mulig. Avstanden fra en landsby til nærmeste brønn kan maksimalt være r. Finn en algoritme for plassering av brønnene med kjøretid $o(n^2)$.

1. Sorter landsbyene basert på distanse fra starten av veien.
2. Gå igjennom landsbyene. For hver landsby som ikke allerede er dekket av en brønn, plasser en brønn i avstand r lengre borte i veien.

Brønnbygging

Oppgave 20: Er sett med landsbyer ligger langs en vei. Skal bygge brønner, men færrest mulig. Avstanden fra en landsby til nærmeste brønn kan maksimalt være r. Finn en algoritme for plassering av brønnene med kjøretid $o(n^2)$.

1. Sorter landsbyene basert på distanse fra starten av veien.
2. Gå igjennom landsbyene. For hver landsby som ikke allerede er dekket av en brønn, plasser en brønn i avstand r lengre borte i veien.

Kjøretid på $\Theta(n \lg n)$.

Brønnbygging

Oppgave 21: Kan nå bare bygge k brønner. Ønsker å finne den minste avstanden d som er slik at avstanden fra hver landsby til nærmeste brønn er maksimalt d. Finn en algoritme for å finne d med kjøretid $o(n^2 + nl)$.

Brønnbygging

Oppgave 21: Kan nå bare bygge k brønner. Ønsker å finne den minste avstanden d som er slik at avstanden fra hver landsby til nærmeste brønn er maksimalt d. Finn en algoritme for å finne d med kjøretid $o(n^2 + nl)$.

1. Sorter landsbyene basert på distanse fra starten av veien.
2. Binærsøk etter d mellom 0 og lengden på veien. Bruk steg 2 fra oppgave 20 til å sjekke om det er mulig med en gitt d.

Brønnbygging

Oppgave 21: Kan nå bare bygge k brønner. Ønsker å finne den minste avstanden d som er slik at avstanden fra hver landsby til nærmeste brønn er maksimalt d. Finn en algoritme for å finne d med kjøretid $o(n^2 + nl)$.

1. Sorter landsbyene basert på distanse fra starten av veien.
2. Binærsøk etter d mellom 0 og lengden på veien. Bruk steg 2 fra oppgave 20 til å sjekke om det er mulig med en gitt d.

Kjøretid i verste tilfellet på $\Theta(n \lg n + n \lg l)$