16.6 4

$R(a,b) \quad S(b,c) \quad T(c,d) \quad U(a,d)$

$T(R) = T(U) = 1000$
$T(S) = T(T) = 200$

$V(R,a) = V(R,b) = V(S,b) = V(T,d) = V(U,a) = V(U,d) = 200$
$V(S,c) = V(T,c) = 20$

a) $T(R \bowtie S) = T(R) * T(S) / max[V(R,b), V(S,b)]$
$= 1000 * 200 / max(200,200) = 1000$

$T(R \bowtie T)$ er ikke aktuell (ingen felles attributter)

$T(R \bowtie U) = T(R) * T(U) / max[V(R,a), V(U,a)]$
$= 1000 * 1000 / max(200,200) = 5000$

$T(S \bowtie T) = T(S) * T(T) / max[V(S,c), V(T,c)]$
$= 200 * 200 / max(20,20) = 2000$

$T(S \bowtie U)$ ikke aktuell

$T(T \bowtie U) = T(T) * T(U) / max[V(T,d), V(U,d)]$
$= 200 * 1000 / max(200, 200) = 1000$

En grådig algoritme vil starte med enten $R \bowtie S$ eller $T \bowtie U$, begge med 1000 tupler. Anta $R \bowtie S$.

$T((R \bowtie S) \bowtie T) = 1000 * 200 / max(V(S,c); V(T,c))$
$= 1000 * 200 / max(20,20) = 10000$

$T((R \bowtie S) \bowtie U) = 1000 * 1000 / max(V(R,a), V(U,a))$
$= 1000 * 1000 / max(200,200) = 5000.$

Deretter vil $(R \bowtie S) \bowtie U$ velges, med 5000 tupler.
Totalt $1000 + 5000 = 6000$ tupler til mellomresultatene.

b) Den optimale rekkefølgen er imidlertid $(R \bowtie S) \bowtie (T \bowtie U)$, med totalt $1000 + 1000 = 2000$ tupler i mellomresultatene.

13.3.1

cylinder of request	first time available (ms)
8000	0
48000	1
4000	10
40000	20

Hodet initielt på spor 16000.

Megatron 747

8 plater, 16 overflater

$2^{16} = 65536$ spor/overflate

Gjennomsnittlig $2^8 = 256$ sektorer/spor

$2^{12} = 4096$ B/sektor

7200 rpm

1ms for å starte og stoppe diskhodet

1ms for å passere 4000 sylindre

Gaps opptar 10% av plassen

a) Elevatoralgoritmen:

t = 0: Beveger seg mot spor 8000 (eneste request på dette tidspunktet

$1 + 0,00025 \cdot (16000 - 8000) = 3\,ms$

start og stopp av hodet   1/4000

t = 3: Er kommet til spor 8000 og betjener requesten.
Med tallene i boken tar det

$8,33/2 + 0,13 = 4,3\ ms$

halv rotasjon   lese en blokk

t = 7,3: Request for spor 48000 er eneste nye request.
Går mot denne:

$1 + 0,00025 \cdot (48000 - 8000) = 11\,ms$

t = 18,3: Ferdiggjør requesten:

4,3ms

t = 22,6: Nærmeste av de to gjenværende (4000 og 40000) er 40000. Går mot denne:

$1 + 0,00025 \cdot (48000 - 40000) = 3\,ms$

t = 25,6: Ferdiggjør requesten:

4,3ms

t = 29,9: Spor 4000 gjenstår, går mot denne:

$1 + 0,00025 \cdot (40000 - 4000) = 10\,ms$

t = 39,9: Ferdiggjør requesten:

4,3 ms

t = 44,2: Ferdig.

13.3.1

b) First-come-first-served:

$t=0$: Mot 8000,

$3 + 4{,}3 = 7{,}3$ ms

$t=7{,}3$: Mot 48000,

$11 + 4{,}3 = 15{,}3$ ms

$t=22{,}6$: Mot 4000,

$1 + 0{,}00025 \cdot (48000 - 4000) = 12$ ms

4,3 ms for å behandle

$t=38{,}9$: Mot 40 000

$10 + 4{,}3 = 14{,}3$ ms

$t=53{,}2$: Ferdig.

13.3.2

To speilede disker, den ene leser fra indre halvdel av sylindrene, den andre fra de ytre.

Anta at leseforespørslene er på tilfeldige spor og at det ikke er skriveforespørsler.

7200 rpm ~ 1 rotasjon pr. 8,33 ms
Flytte hodet: 1ms start+stopp) + 1ms pr. 4000 sylindre i forflytning
$65536 = 2^{16}$ spor pr. overflate, 8 plater med 16 overflater
$256 = 2^{8}$ sektorer i snitt pr. spor
$2^{12} = 4096$ bytes pr. sektor
1 blokk = 16384 bytes
gaps mellom sektorene utgjør 10% av sirkelen/sporet, sektorene utgjør 90%.

a) Gjennomsnittlig tid for lesing av en blokk:

Én blokk går over 16384 : 4096 = 4 sektorer
256 sektorer og 256 gaps pr. spor
4 sektorer og de 3 mellomliggende gapene utgjør

$$360 \cdot \frac{90}{100} \cdot \frac{4}{256} + 360 \cdot \frac{10}{100} \cdot \frac{3}{256} = 5,48 \text{ grader av en sirkel}$$

Overføringstiden for blokken er følgelig

$$\frac{5,48}{360} \cdot 8,33 \text{ ms} = 0,13 \text{ millisekunder}$$ (se forøvrig eksempel 13.2 i boka)

Gjennomsnittlig rotasjonsforsinkelse er 1/2 runde, dvs.

$$\frac{8,33}{2} \text{ ms} = 4,17 \text{ ms}$$

Gjennomsnittlig må hodet passere 1/3 av radius for å finne rett spor; tiden dette tar er (med start og stopp av hodet på 1 ms)

$$1 + \frac{65536}{2} \cdot \frac{1}{3} \cdot \frac{1}{4000} \text{ ms} = 3,73 \text{ ms}$$

For hver disk er altså gjennomsnittlig totaltid

$$3,73 + 4,17 + 0,13 \text{ ms} = 8,03 \text{ ms}$$ dvs. 1000/8,03 = 124 blokker pr. sekund (124,5..)

Siden de to diskene jobber i parallell, overføre de to blokker på denne tiden, så totalt

$$2000/8,03 = 249$$

diskblokker pr. sekund.

b) Gjennomsnittlig tid for hver disk hvis alle spor gjennomsøkes på begge, er (iflg. eksempel i boka, regnet ut som over) 10,76 ms, så totalt

$$2 \cdot 1000/10,76 = 185 \text{ diskblokker pr. sekund}$$

a) gir en økning på 249/185 = 1,35, dvs. 35% mer enn b).

13.3.2

c) Kan ikke velge fra hvilken disk blokkene skal leses, så mindre fleksibilitet for scheduleren. Hvis det ikke er fullstendig randomisert hvor lesingen skjer, er man fort i en situasjon hvor én disk står for mesteparten av aksessene, og da er resultatet verre enn om begge disker leses fra alle spor. Da overføres (bare fra én disk)

124 blokker pr. sekund

, i forhold til b) er dette $185/124 = 1{,}49$, dvs. nesten 50% dårligere enn b).

13.4.5

RAID 4

a)

$$
\begin{array}{r}
01010110 \\
\oplus\ 11000000 \\
\oplus\ 00101011 \\
\oplus\ 10111011 \\
\hline
=\ 00000110
\end{array}
$$

b)

$$
\begin{array}{r}
b)\ 11110000 \\
\oplus\ 11111000 \\
\oplus\ 00111100 \\
\oplus\ 01000001 \\
\hline
=\ 01110101
\end{array}
$$

Oppgave 2

$F = \{ DE \to AD, AB \to BC, BC \to CD, CD \to ABE \}$

1. Minimal overdekning:

a Splitter høyresidene (og fjerner trivielle FDer)

$DE \to A$

triviell $\cancel{DE \to D}$

-||- $\cancel{AB \to B}$

$AB \to C$

-||- $\cancel{BC \to C}$

$BC \to D$

$CD \to A$

$CD \to B$

$CD \to E$

b Minimale venstresider:

Alle venstresider er minimale:
siden det ikke er noen FD med
bare ett attributt i venstresiden.
F.eks. for $DE \to A$:

$D^+ = D$, $E^+ = E$,

så verken $D \to A$ eller $E \to A$
holder.

c Fjern overflødige FDer:

$DE^+ = DE$ i $F - \{DE \to A\}$

$AB^+ = AB$ i $F - \{AB \to C\}$

$BC^+ = BC$ i $F - \{BC \to D\}$

$CD^+ = CDBEA$ i $F - \{CD \to A\}$ — $CD \to A$ er overflødig

$CD^+ = CDAE$ i $F - \{CD \to B\}$

$CD^+ = CDAB$ i $F - \{CD \to E\}$

Resultat:

$DE \to A$

$AB \to C$

$BC \to D$

$CD \to B$

$CD \to E$

2.

Elementær FD: $X \to A$ der

- A er <u>ett</u> attributt
- $X \to A$ er ikketriviell (dvs. $A \notin X$)
- X minimal (dvs. har ikke $Y \to A$ for noen ekte delmengde Y av X)

Alle FDene i den minimale overdekningen er elementære:

$DE \to A$

$AB \to C$

$BC \to D$

$CD \to B$

$CD \to E$

($A^+ = A$, $B^+ = B$, $C^+ = C$, $D^+ = D$, $E^+ = E$, så ingen venstreside kan gjøres mindre og fortsatt utgjøre en FD.)

3. Kandidatnøkler: Enhver kandidatnøkkel må bestå av minst to attributter siden alle venstresider har to attributter.

$DE^+ = DEA$

$AB^+ = ABCDE$ Kandidatnøkkel: AB

$BC^+ = BCDEA$ Kandidatnøkkel: BC

$CD^+ = CDBEA$ Kandidatnøkkel: CD

$AC^+ = AC$, $AD^+ = AD$, $AE^+ = AE$, $BD^+ = BD$, $BE^+ = BE$, $CE^+ = CE$, $ACE^+ = ACE$

Så

$AB \to C$, $BC \to D$, $CD \to B$, $CD \to E$

oppfyller kravet til BCNF. For

$DE \to A$

er venstresiden <u>ikke</u> en supernøkkel, så kravet til BCNF er brutt. Høyreside er et nøkkelattributt (nøkkelattributtene er $ABCD$), så kravet til 3NF er oppfylt. Nå er $AB \to C$ en elementær FD, så AB er en elementær kandidatnøkkel. Følgelig oppfyller $DE \to A$ kravet til EKNF.

Totalt er R derfor på EKNF.

4. Tapsfri dekomposisjon til BCNF:

$$R(A,B,C,D,E)$$

DE → A bryter BCNF

$$R1(A,D,E) \qquad R2(B,C,D,E)$$

5. Det fins ingen FD-bevarende dekomposisjon til BCNF.
Hvis vi bruker algoritmen for FD-bevarende dekomposisjon (til EKNF eller bedre), får vi følgende relasjoner:

$R_{AB} = \{ \underline{A,B}, C \}$ fra $AB \to C$

$R_{BC} = \{ \underline{B,C}, D \}$ fra $BC \to D$

$R_{CD} = \{ B, \underline{C,D}, E \}$ fra $CD \to B$, $CD \to E$. R_{CD} inneholder også FDen $BC \to D$.

$R_{DE} = \{ A, \underline{D,E} \}$ fra $DE \to A$

$BC \to D$ i R_{CD} bryter BCNF. Så denne er på EKNF, men ikke BCNF. Hvis det hadde vært mulig å dekomponere tapsfritt og FD-bevarende til BCNF, ville denne algoritmen som vi her har brukt, gitt en slik dekomposisjon.