Løsningsforslag Eksamen i fag
SIF8010 Algoritmer og Datastrukturer
Fredag 11. desember 1998

1 Oppgave 1

$T(1) = K_1 + K_2$
$T(N) = T(5^m) = K_1 + K_2 N + T(\frac{N}{5})$
$T(N) = K_1(\log_5 N + 1) + K_2(\frac{5^{N-1}}{4})$
$T(N) = O(N)$

2 Oppgave 2

Gjennomsnitts-reise/element: $\frac{1}{n}\sum_j(\frac{1}{n}\sum_i |i-j|) = \frac{n}{3} - \frac{1}{3n}$
Total flytting: $n(\frac{n}{3} - \frac{1}{3n}) = \frac{n^2}{3} - \frac{1}{3}$
Nedre grense for arbeid er $\Omega(n^2)$
Under disse antagelsene er utsagnet riktig.

3 Oppgave 3

(a)
Ja, fordi alle verdier i et intervall [x,y] pr. def. må være der.
(b)
a = b : Trivielt (alle verdier like).
$a < b$: Binær søking: z må befinne seg i en av $^1/_2$-delene. (z kan være i begge halvdelene, men det holder å søke i en av disse.)
$a > b$: Tilsvarende binær søk, speilvendt.
(c)
$\lceil log_2 n \rceil$ (Binær-søk)

4 Oppgave 4

(a)
32
(b)

1. Sorter G topologisk.
2. Gjør følgende i topologisk rekkefølge:
   Antall veier til node i, $|vei_i| = \sum_{j \in i's\, forgjengere} |vei_j|$

Uten topologisk sortering er det vanskelig å finne i's forgjengere.
(c)
Fra (b) har vi:

1. $\Theta(|V| + |E|)$
2. $\Theta(|V| + |E|)$

Tidskompleksitet: $2\Theta(|V| + |E|) = \Theta(|V| + |E|)$
(d)
Noen eksempeler kan være:

- Lage synonymer for søking $\rightarrow$ Skjult søk på alle, samtidig.
- "Normalisering" av beslektede navn: Jonsen,V12 = Johannesson. (Effektivt til variant-søk)

5 Oppgave 5

(a)
Kan åpenbart i $O(|V||E|)$ tid sjekke om en tilfeldig 3-fordeling gir konflikter/er konfliktfri.
(b)
Fordi F(G,2) lett løses i $O(|V|+|E|)$ (polynomisk!) tid: Farg annenhver påstøtt node svart/hvit for:

1. enten oppdage odde loop
2. eller farge hele G.

6 Oppgave 6 - Øvingsrelaterte oppgaver

(a)

- små datamengder
- stor orden (nesten sortert), eller kun putte inn ett nytt element
- enkelhet, i lite kritisk applikasjon

(b)

- store datamengder

- uavhengig av forhåndsordning
- kritisk applikasjon med <u>stor</u> datamengde, slik at konstantledd-virkningen oppheves av forholdet.

(c)
"Vektortre" : Gjennom øving 1 ble det lagt opp til at det i øving 3 skulle brukes Heapsort.
(d)
$O(n \log_2 n)$, takket være Heapsort
(e)
Uten Heapsort, ingen log n - faktor.
(f)
Ved bruk av f.eks. Liste istedet for Heap blir tidskompleksiteten $O(n^2)$
(g)
Pi-tabell: π = [0, 0, 0, 1, 2, 0, 0, 1, 2]
(h)

```
TGAGACGACGATAGA   (Søketekst)
GACGATAGA         (Start)
 GACGATAGA
   GACGATAGA
      GACGATAGA   (Slutt - tekst funnet)
```