45011 Algoritmer og datastrukturer
Løsningsforslag eksamen 11. januar 1994

Oppgave 1

a	$O(n)$	$T(n) = 1 + 1 + \cdots + 1 = O(n)$ (iterasjon)
b	$O(n \lg^3 n)$	Master-teoremet, tilfelle 2. (Spesialtilfelle, se Ex. 4.4-2)
c	$O(n^2)$	Master-teoremet, tilfelle 3
d	$O(n^{\log_2 3})$	Master-teoremet, tilfelle 1
e	$O(n)$	v/ substitusjon. Viser $T(n) \leq cn$ ved induksjon. For store n er $n/2 + \sqrt{n} \to n/2$, så $O(n)$ er en naturlig gjetning.

Oppgave 2

- Utfør $A'[i] := A[i] - 1$ for alle i for å verdiene i området $[0, n^3 - 1]$. Dette tar $O(n)$ tid.
- Behandle $\{A'[i]\}$ som 3-sifrede tall $\{a_1 a_2 a_3\}$, der $a_i \in [0, n-1]$, dvs radix.
- Sorter 3 runder med radix-sortering. F.eks. ved $2*$ ”vektor av lister”, hver vektor med lengde n. $3O(n)$ tid.
- Gjenopprett $A[i] = A'[i] + 1$. $O(n)$ tid.

Dette tar totalt $O(n)$ tid.

Overser her (for avansert) at lengden til et tall $A[i]$ er $O(\lg n)$. Dette kan diskuteres.

Oppgave 3

a) Ja: Bellman-Fords algoritme tolererer negative linjelengder såfremt ingen sykel med negativ lengde finnes. Løsningen blir da fortegn-snuing, deretter krav om at ingen negativ sykel finnes etter snuingen. Minstekrav til G: Ingen negative sykler etter fortegn-snuing.

b) Dette er øving 25.2-4.

Init-single-source:

INIT-SINGLE-SOURCE-REL(G, s)
for each $v \in V[G]$ **do**
    $d[v] \leftarrow -\infty$
    $\pi[v] \leftarrow$ **nil**
$d[s] \leftarrow 1$

Extract-Min: Erstattes med EXTRACT-MAX

Relax: Kun operatorendring: '+' byttes ut med '$\cdot$'

RELAX-RELIABILITY(u, v, r)
if $d[v] < d[u] \odot r(u, v)$ **then**
    $d[v] \leftarrow d[u] \odot r(u, v)$
    $\pi[v] \leftarrow u$

c) $w(u, v) = -\lg(r(u, v))$ (positive tall fordi $r(u, v) \in [0, 1]$)
Søker maksimering av $\prod_{(u,v)\in \text{sti}} r(u, v)$. Dette er ekvivalent med å maksimere $\lg(\prod r(u, v)) = \sum \lg(r(u, v))$. Dette er igjen ekvivalent med å minimere $\sum -\lg(r(u, v))$. Det er greit at $\log 0 = -\infty$, fordi dette garanterer at usikre veier blir unngått hvis alternativ finnes.

Oppgave 4

Modell: Vi lager en hjelpekant fra t til s.

Algoritme: Basert på binærsøk.

1. Sett:
   $M := \min[\sum(\text{kapasitet ut fra } s), \sum(\text{kapasitet inn til } t)]$; (øvre grense, lett å finne)
   $b := M$
   $a := b/2$

2. Finner **A** gyldig flyt med verdi $F \in [a, b]$?
   Ja: $a := (F + b)/2$
   Nei: $b := a$; $a := (F + a)/2$ (gammel F hittil best)

3. Fortsett med 1 hvis og bare hvis $a < b$, ellers retur med siste F-verdi (=a=b) = Flytmaks-verdien.

Kompleksitet: $O((|E| + |V|) \cdot \lg M)$

Oppgave 5

Vi lager en ny post i typen NODE: nodestatus : integer;
nodestatus = 0 initielt.
1 noder på stakk, ikke behandlet.
2 alle etterfølgende noder undersøkt, kan ikke inngå i sykel.

```
Function SYKELFRI(startnode : NodeRef):Boolean;
var sykel : Boolean;
    Procedure NODEVISITT(nestenode:NodeRef);
    var nabokant : KantRef;
    begin
        if nestenode^.nodestatus = 1 then
            sykel :=true;
        else
        begin
            if nestenode^.nodestatus = 0 then
            begin
                nestenode^.nodestatus := 1;
                nabokant := nestenode^.fuk;
                while nabokant ≠ nil and not sykel then
                begin
                    NODEVISITT(nabokant^.enode);
                    nabokant := nabokant^.nuk;
                end;
                nestenode^.nodestatus := 2;
            end;
        end;
    end;
begin
sykel := false;
NODEVISITT(startnode);
SYKELFRI := not sykel;
end;
```