

## 582359 Algoritmit ongelmanratkaisussa (kevät 2013)

### Viikon 1 tehtävät (18.1.)

1. Laske seuraavat bittioperaatiot ensin käsin. Varmista sitten ohjelmalla, että tulokset ovat oikein.

- (a)  $123 \ \& \ 321$  (and-operaatio)
- (b)  $123 \ | \ 321$  (or-operaatio)
- (c)  $123 \ ^ \ 321$  (xor-operaatio)
- (d)  $23 \ \ll \ 3$  (bittisiirto vasemmalle)
- (e)  $157 \ \gg \ 4$  (bittisiirto oikealle)
- (f)  $\sim 123$  (not-operaatio)

2. Perustele seuraavissa kohdissa, miksi vasemmalla ja oikealla olevat koodit toimivat samalla tavalla. Mitä hyviä ja huonoja puolia bittioperaatioiden käytössä on kussakin tilanteessa?

(a) Parillisuuden tarkastus:

```
if (x % 2 == 0) {                               if (x & 1 == 0) {
```

(b) Muuttujan nollaus:

```
x = 0;                                           x ^= x;
```

(c) Kerto- ja jakolaskut:

```
x = (int)Math.pow(2, 20);                       x = 1 << 20;
a = 256 * b;                                     a = b << 8;
c = d / 2;                                       c = d >> 1;
e = 24 * f;                                       e = (f << 4) + (f << 3);
```

(d) Muuttujien a ja b arvojen vaihtaminen:

```
apu = a;                                         a = a ^ b;
a = b;                                           b = a ^ b;
b = apu;                                         a = a ^ b;
```

3. Lukujen  $1 \dots n$  osajoukko voidaan esittää yhtenä kokonaislukuna *bittivektorina*. Jos luku  $k$  kuuluu osajoukkoon, niin bittivektorin  $k$ . bitti lopusta alkaen on 1, ja muuten bitti on 0. Esimerkiksi osajoukkoa  $\{1, 3, 4, 8\}$  vastaa bittivektori 10001101 eli kokonaisluku 141.

Tiedostossa `Osajoukko.java` on osajoukon toteutus boolean-taulukkona. Muuta ohjelmaa käyttämään bittivektoria, jolloin osajoukon tallennukseen riittää yksi int-muuttuja.

4. Syötteenä annetaan positiivinen kokonaisluku  $n$  sekä  $n - 1$  eri kokonaislukua väliltä  $1 \dots n$ . Tehtävänä on etsiä puuttuva luku. Esimerkiksi jos  $n = 5$  ja luvut ovat 3, 1, 5 ja 4, niin puuttuva luku on 2.

Esitä tehtävään ajassa  $O(n)$  ja tilassa  $O(1)$  toimiva ratkaisu, joka hyödyntää xor-operaatiota.

5. Funktio  $Y(n)$  vastaa lukujen  $1 \dots n$  ykkösbittien yhteismäärää. Esimerkiksi  $Y(5) = 7$ , koska lukujen  $1 \dots 5$  bittiesitykset ovat 1, 10, 11, 100 ja 101 ja niissä on yhteensä 7 ykkösbittia.

Tee ohjelma, joka laskee tehokkaasti arvon  $Y(12345678987654321)$ .

6. (a) Käytettävissä ovat luvut  $\{1, 2, 4, 8, 16, 32, \dots\}$  eli muotoa  $2^k$  olevat luvut. Osoita, että mikä tahansa positiivinen kokonaisluku voidaan esittää näiden lukujen osajoukon summana.

Esimerkiksi  $185 = 1 + 8 + 16 + 32 + 128$ .

(b) Osoita sama, kun luvut ovat  $\{1, -2, 4, -8, 16, -32, \dots\}$  eli muotoa  $2^k(-1)^k$ .

Esimerkiksi  $185 = 1 - 8 + 64 - 128 + 256$ .