

Predstavljanje skupova nizom
bita

Problem

- Skupove možemo predstaviti na više načina. Pokazaćemo način predstavljanja za skupove koji imaju ne više od 64 elementa. Ovaj metod je veoma brz, jer skupove predstavljamo kao cijele brojeve

Rješenje

- Neka je A proizvoljni podskup skupa $\{0, 1, \dots, N - 1\}$. Skup A možemo predstaviti kao binarni broj dužine N bita, gdje bit na poziciji i (koji predstavlja 2^i) jednak 1 ako $i \in A$ i jednak je 0 ako $i \notin A$. Većina programskih jezika podržavaju cijele brojeva sa najviše 64 bita, pa jednim cijelim brojem možemo predstaviti skup od najviše 64 elementa. U datim primjerima korišćeni su 32-bitni cijeli brojevi.

Primjer 1

- $N=16$, $A=\{2, 3, 5, 7\}$
- Skup A možemo predstaviti binarnim nizom dužine 16:

ind	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0

- $(0000000010101100)_2 = 2^7 + 2^5 + 2^3 + 2^2 = 128 + 32 + 8 + 4 = 172$
- Skup A kodiramo cijelim brojem 172

Primjer 2

- $N=16$, $B=\{0, 2, 4, 7\}$
- Skup B možemo predstaviti binarnim nizom dužine 16:

ind	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0

- $(0000000010010101)_2 = 2^7 + 2^4 + 2^2 + 2^0 = 128 + 16 + 4 + 1 = 149$
- Skup B kodiramo cijelim brojem 149

Presjek i unija

- $A \cup B = \{0, 2, 3, 4, 5, 7\}$ – bitwise A | B
- $A \cap B = \{2, 7\}$ – bitwise A & B

Ind	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0
B	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
$A \cup B$	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0
$A \cap B$	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

- $A|B = 189$
- $A \& B = 132$

Ostale operacije (1)

- Razlika skupova A i B :
 $A \& \sim B$
- Da li je skup prazan:
 $A \neq 0$
- Skup $\{0, \dots, x - 1\}$:
 $(1 \ll x) - 1$
- Da li x pripada skupu A:
 $(A \& (1 \ll x)) \neq 0$

Ostale operacije (2)

- Negacija skupa (Komplement)

$$((1 \ll N) - 1) \& \sim A$$

- Dodati x u skup A (čak i ako je x već pripada skupu A):

$$A |= 1 \ll x;$$

- Ukloniti x iz A (čak i ako x ne pripada skupu A):

$$A \&= \sim(1 \ll x);$$

Ostale operacije (3)

- Da li A ima tačno jedan element:
 $(A > 0) \ \&\& \ (A \ \& \ (A - 1)) \ == \ 0$
- Ukloniti najmanji element iz A:
 $A = A \ \& \ (A - 1)$
- Ukloniti sve elemente osim najmanjeg:
 $A = A \ \& \ \sim(A - 1)$

Ostale operacije (4)

- Broj elemenata skupa A:

// GCC:

```
cnt = __builtin_popcount(A);
```

// Java:

```
cnt = Integer.bitCount(A);
```

// Portable:

```
cnt = (A & 0x55555555) + ((A >> 1) & 0x55555555);
```

```
cnt = (cnt & 0x33333333) + ((cnt >> 2) & 0x33333333);
```

```
cnt = (cnt & 0x0F0F0F0F) + ((cnt >> 4) & 0x0F0F0F0F);
```

```
cnt = (cnt & 0x00FF00FF) + ((cnt >> 8) & 0x00FF00FF);
```

```
count = (cnt & 0x0000FFFF) + (cnt >> 16);
```

Ostale operacije (5)

- Indeks najmanjeg elementa

// GCC:

```
low = __builtin_ctz(A);
```

// Java:

```
low = Integer.numberOfTrailingZeros(A);
```

// Portable:

```
low = 0; lowbit = A & ~(A - 1);
```

```
if ((tmp & 0xAAAAAAAA) != 0) low += 1;
```

```
if ((tmp & 0xCCCCCCCC) != 0) low += 2;
```

```
if ((tmp & 0xF0F0F0F0) != 0) low += 4;
```

```
if ((tmp & 0xFF00FF00) != 0) low += 8;
```

```
if ((tmp & 0xFFFF0000) != 0) low += 16;
```