

Struttura dati SET implementata come Vettore Booleano

bool[] V

int $size$

int dim

SET **Set**(**int** m)

 SET $t \leftarrow$ new SET

$t.size \leftarrow 0$

$t.dim \leftarrow m$

$t.V \leftarrow [\text{falso}] * m$

 ritorna t

SET **contains**(**int** x)

 se $1 \leq x \leq dim$ allora

 ritorna $V[x]$

 allora

 ritorna falso

int $size$

 ritorna $size$

insert(**int** x)

 se $1 \leq x \leq dim$ allora

 se not $V[x]$ allora

$size++$

$V[x] \leftarrow \text{vero}$

remove(**int** x)

 se $1 \leq x \leq dim$ allora

 se $V[x]$ allora

$size--$

$V[x] \leftarrow \text{falso}$

SET **union**(SET A , SET B)

// crea un insieme della capacità max

SET $C \leftarrow \text{Set}(\max(A.dim, A.dim))$

// inserisci gli elementi di A

da $i \leftarrow 1$ fino a $A.dim$ fai

 se $A.contains(i)$ allora

$C.insert(i)$

// inserisci gli elementi di B

da $i \leftarrow 1$ fino a $B.dim$ fai

 se $A.contains(i)$ allora

$C.insert(i)$

SET **intersection**(SET A , SET B)

// crea un insieme della capacità min

SET $C \leftarrow \text{Set}(\min(A.dim, A.dim))$

da $i \leftarrow 1$ fino a $\min(A.dim, A.dim)$ fai

 // se è contenuto in entrambi

 se $A.contains(i)$ and $B.contains(i)$ allora

$C.insert(i)$ // aggiungilo

SET **difference**(SET A , SET B)

SET $C \leftarrow \text{Set}(A.dim)$

da $i \leftarrow 1$ fino a $A.dim$ fai

 // se è contenuto A e non in B

 se $A.contains(i)$ and not $B.contains(i)$ allora

$C.insert(i)$ // aggiungilo