

# Capitolo 4

## Lezione 28 marzo 2019

### 4.1 Insiemi

Si supponga di rappresentare gli insiemi di interi utilizzando degli array per memorizzare gli elementi presenti in ciascun insieme nel seguente modo:

- la prima posizione dell'array contiene il numero di elementi presenti. Un insieme di 5 elementi viene rappresentato utilizzando le prime 6 posizioni dell'array
- i nuovi elementi vengono inseriti dopo l'ultimo elemento già inserito
- se un elemento è già presente non viene inserito (si tratta di un insieme), quindi tutti gli elementi presenti nel array (nelle posizioni successive a quella utilizzata per indicare la cardinalità) devono essere distinti
- le operazioni sugli insiemi devono ritornare degli insiemi validi (senza duplicati)

```
#define MAX_ELEMENTI 1000
typedef int insieme [MAX_ELEMENTI+1];
```

Implementare le seguenti funzioni C:

```
/* Inizializza l'insieme A impostando il numero di elementi a 0 */
void inizializza_insieme(insieme A);

/* Inserisce l'elemento x nell'insieme A
restituisce 1 in caso di successo, 0 se non c'è spazio per nuovi elementi */
int inserisci_elemento_insieme(insieme A, int x);

/* Elimina elemento x dall'insieme A
restituisce 1 in caso di successo, 0 se x è assente */
int inserisci_elemento_insieme(insieme A, int x);

/* Restituisce 1 se l'insieme A contiene x, 0 altrimenti */
int insieme_contiene(insieme A, int x);

/* inserisce nell'insieme risultato gli elementi comuni ad A e B */
int intersezione_insiemi(insieme A, insieme B, insieme risultato);

/* inserisce (una sola volta) nell'insieme risultato gli elementi contenuti in A o B (o entrambi,
non esclusivo) */
int intersezione_insiemi(insieme A, insieme B, insieme risultato);

/* inserisce nell'insieme risultato gli elementi contenuti in A ma non in B */
int differenza_insiemi(insieme A, insieme B, insieme risultato);
```

## 4.2 Insiemi ordinati

Si supponga di rappresentare gli insiemi ordinati di interi utilizzando degli array per memorizzare gli elementi presenti in ciascun insieme nel seguente modo:

- la prima posizione dell'array contiene il numero di elementi presenti. Un insieme di 5 elementi viene rappresentato utilizzando le prime 6 posizioni dell'array
- i nuovi elementi vengono inseriti in modo tale che l'ordine crescente degli elementi sia rispettato
- se un elemento è già presente non viene inserito (si tratta di un insieme)
- tutti gli elementi presenti nel array (nelle posizioni successive a quella utilizzata per indicare la cardinalità) devono essere distinti ed in ordine crescente
- le operazioni sugli insiemi devono ritornare degli insiemi ordinati validi (elementi in ordine crescente e senza duplicati)

```
#define MAXELEMENTI 1000
typedef int insieme_ordinato [MAXELEMENTI+1];
```

Implementare le seguenti funzioni C, rispettando l'ordine degli elementi negli insiemi risultanti e sfruttando l'ordine degli elementi per implementare le operazioni in modo efficiente:

```
/* Inizializza l'insieme ordinato A impostando il numero di elementi a 0 */
void inizializza_insieme_ordinato(insieme_ordinato A);

/* Inserisce l'elemento x nell'insieme ordinato A
   restituisce 1 in caso di successo, 0 se non c'è spazio per nuovi elementi */
int inserisci_elemento_insieme_ordinato(insieme_ordinato A, int x);

/* Elimina elemento x dall'insieme ordinato A
   restituisce 1 in caso di successo, 0 se x è assente */
int inserisci_elemento_insieme_ordinato(insieme_ordinato A, int x);

/* Restituisce 1 se l'insieme ordinato A contiene x, 0 altrimenti */
int insieme_ordinato_contiene(insieme_ordinato A, int x);

/* inserisce nell'insieme risultato gli elementi comuni ad A e B */
int intersezione_insiemi_ordinati(insieme_ordinato A, insieme_ordinato B, insieme_ordinato
    risultato);

/* inserisce (una sola volta) nell'insieme risultato gli elementi contenuti in A o B (o entrambi,
   non esclusivo) */
int intersezione_insiemi_ordinati(insieme_ordinato A, insieme_ordinato B, insieme_ordinato
    risultato);

/* inserisce nell'insieme risultato gli elementi contenuti in A ma non in B */
int differenza_insiemi_ordinati(insieme_ordinato A, insieme_ordinato B, insieme_ordinato risultato
    );
```