

# **Corso di Laurea in INFORMATICA**

**a.a. 2012-2013**

## **Algoritmi e Strutture Dati MODULO 6**

### **Insiemi**

Specifiche, rappresentazione e confronto tra realizzazioni alternative.

Questi lucidi sono stati preparati da per uso didattico. Essi contengono materiale originale di proprietà dell'Università degli Studi di Bari e/o figure di proprietà di altri autori, società e organizzazioni di cui e' riportato il riferimento. Tutto o parte del materiale può essere fotocopiato per uso personale o didattico ma non può essere distribuito per uso commerciale. Qualunque altro uso richiede una specifica autorizzazione da parte dell'Università degli Studi di Bari e degli altri autori coinvolti.



# Problema

- Progettare un programma che legga il testo di un libro (ad esempio, *I promessi Sposi*) consenta all'utente di dare in input una o più parole e restituisca come risultato l'indicazione che quelle parole siano o meno contenute nel testo.
  - Avrebbe senso usare una lista per memorizzare il testo?
  - Sarebbe più semplice risolvere il problema se disponessimo dell'insieme dei termini (glossario) contenuti nel testo?
- Si noti che per risolvere questo problema non abbiamo bisogno di tutte le operazioni solitamente consentite su una lista ma soltanto delle operazioni di inserimento o ricerca nel glossario .
  - L'ordine col quale sono gli elementi nel glossario può aiutare nelle operazioni?

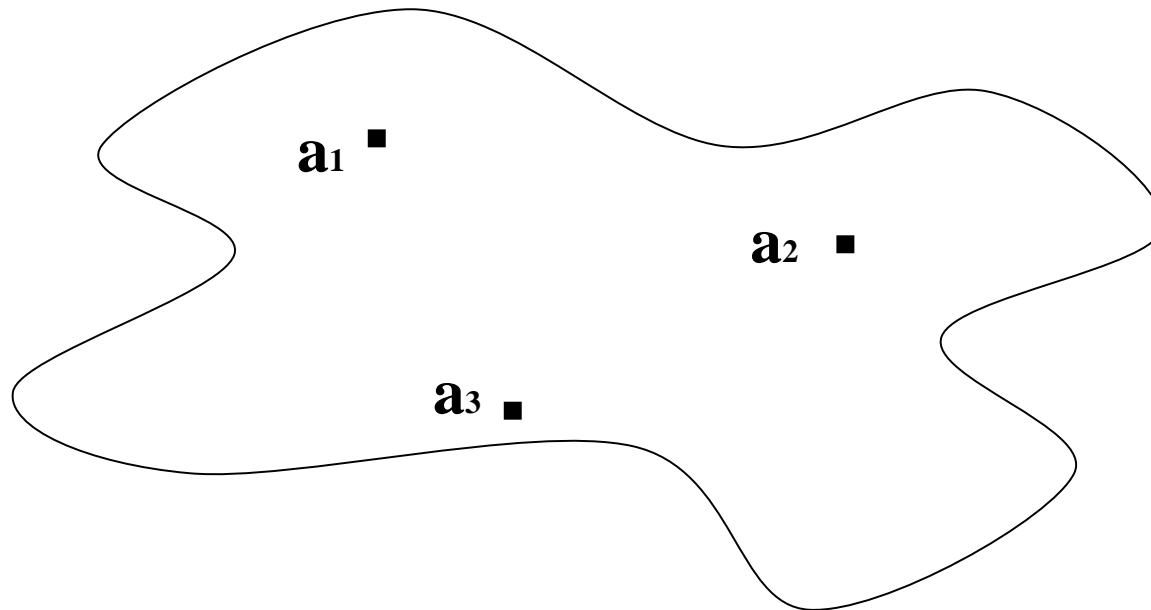
**La struttura utile è l'insieme**



# INSIEMI

UN INSIEME È UNA COLLEZIONE (O FAMIGLIA) DI ELEMENTI (COMPONENTI O MEMBRI) DI TIPO OMOGENEO. A DIFFERENZA DELLE LISTE GLI ELEMENTI **NON** SONO CARATTERIZZATI DA UNA POSIZIONE NÉ POSSONO APPARIRE PIÙ DI UNA VOLTA.

SOLITAMENTE SONO RAPPRESENTATI GRAFICAMENTE



IN MATEMATICA SONO DEFINITI **ESTENSIONALMENTE**  
 **$A = \{ \text{GIALLO, ROSSO, BLU} \}$**

OPPURE **INTENSIONALMENTE** ATTRAVERSO LE PROPRIETÀ CHE DEVONO AVERE I COMPONENTI

**$B = \{ \text{ELEMENTI NEL COLLEGIO BA19 NEL 1994} \}$**

**$C = \{ \text{NUMERI REALI COMPRESI TRA 0 E 1} \}$**

IN INFORMATICA CI RIFERIAMO AL MODO ESTENSIONALE

IL NUMERO DI ELEMENTI  $|A|$  È DETTO CARDINALITÀ E RAPPRESENTA LA DIMENSIONE DELL'INSIEME

**$|A| = 3$**

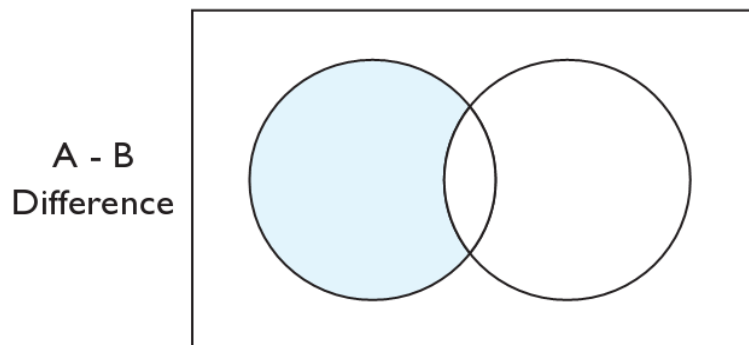
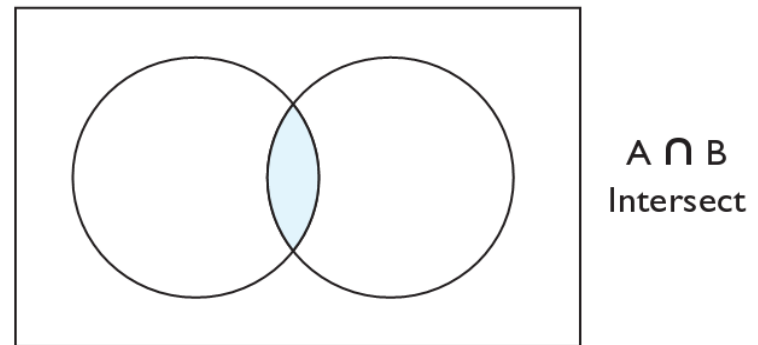
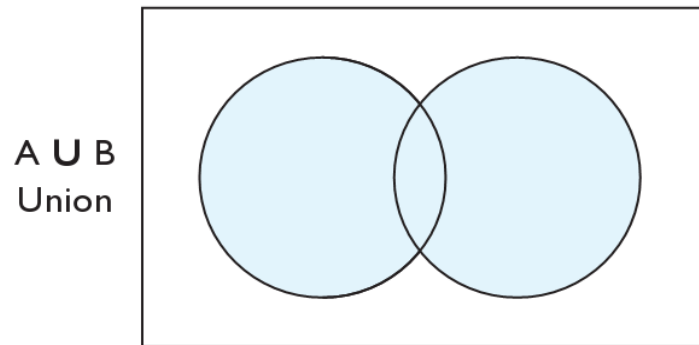
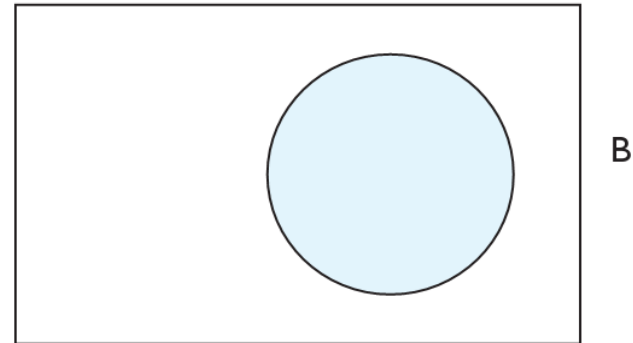
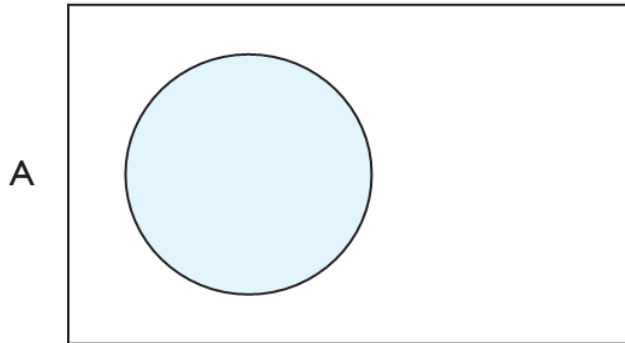
**$|B|$  È FINITA**

**$|C|$  È INFINITA**

LA RELAZIONE FONDAMENTALE È QUELLA DI APPARTENENZA  $x \in A$ , DA CUI DERIVA L'INCLUSIONE  $B \subseteq A$ .



# Operazioni su insiemi



**OPERAZIONI PRINCIPALI SONO :**

<b>UNIONE</b>	$A \cup B$
<b>INTERSEZIONE</b>	$A \cap B$
<b>DIFFERENZA</b>	$A - B$



## SPECIFICA

**TIPI : INSIEME (FAMIGLIA DI INSIEMI COSTITUITA DA ELEMENTI DI TIPO TIPOELEM), BOOLEAN (INSIEME DEI VALORI VERITÀ), TIPOELEM**

**OPERATORI :**

**CREAINSIEME:**  $() \rightarrow \text{INSIEME}$

**CREAINSIEME** = A

POST : A =  $\emptyset$

**INSIEMEVUOTO:**  $( \text{INSIEME} ) \rightarrow \text{BOOLEAN}$

**INSIEMEVUOTO(A)** = b

POST : b = VERO

b = FALSO

SE A =  $\emptyset$

ALTRIMENTI

**APPARTIENE:**  $( \text{TIPOELEM} , \text{INSIEME} ) \rightarrow \text{BOOLEAN}$

**APPARTIENE (x , A)** = b

POST : b = VERO

b = FALSO

SE  $x \in A$

ALTRIMENTI



**INSERISCI:** (TIPOELEM , INSIEME )  $\rightarrow$  INSIEME

**INSERISCI** (x , A ) = A'

PRE :  $x \notin A$  (OPPURE SENZA PRECONDIZIONE)

POST :  $A' = A \cup \{x\}$  (SE  $x \in A$  ,  $A \equiv A'$  )

**CANCELLA:** (TIPOELEM , INSIEME )  $\rightarrow$  INSIEME

**CANCELLA** (x , A ) = A'

PRE :  $x \in A$  (OPPURE SENZA PRECONDIZIONE)

POST :  $A' = A - \{x\}$  (SE  $x \notin A$  ,  $A \equiv A'$  )



**UNIONE:** (INSIEME, INSIEME)  $\rightarrow$  INSIEME

**UNIONE ( A , B ) = C**

**POST :**  $C = A \cup B$

**INTERSEZIONE:** (INSIEME, INSIEME)  $\rightarrow$  INSIEME

**INTERSEZIONE ( A , B ) = C**

**POST :**  $C = A \cap B$

**DIFFERENZA:** (INSIEME, INSIEME)  $\rightarrow$  INSIEME

**DIFFERENZA ( A , B ) = C**

**POST :**  $C = A - B$





**L'INSIEME È UN DATO PRIMITIVO IN PASCAL.**

**INFATTI, E' DISPONIBILE IL SET**

**(TYPE <IDENT.> = SET OF <TIPO-base>)**

**GLI OPERATORI:**

**APPARTENENZA**

**INCLUSIONE**

**UNIONE**

**INTERSEZIONE**

**DIFFERENZA**

**SONO INDICATI**

**IN**

**<=**

**+**

**\***

**-**

**PER GLI ASSEGNAMENTI**

**C := [ ] ;            B := B + [SUCC(TIPO SCALARE )];**

**D := [ 'd' , 'f' , 'g' ];            C := D - [ch]**



# REALIZZAZIONE CON VETTORE BOOLEANO

PER LINGUAGGI CHE NON DISPONGONO  
DEL TIPO INSIEME, È POSSIBILE  
RAPPRESENTARE UN INSIEME  $A$ , I CUI  
ELEMENTI SIANO, AD ESEMPIO, INTERI  
 $\in [1, N]$ , ATTRAVERSO UN VETTORE  
BOOLEANO DI  $N$  BIT, IL CUI  $K$ -esimo  
VALORE SARÀ “VERO” SE  $K \in A$  E  
“FALSO” SE  $K \notin A$ . (VETTORE  
CARATTERISTICO)

AD ESEMPIO L'INSIEME  
(1,3,4,8,10,12)



1	T
2	F
3	T
4	T
5	F
6	F
7	F
8	T
9	F
10	T
11	F
12	T



UN'ALTRA POSSIBILE **REALIZZAZIONE** SI AVVALE DI  
UNA **LISTA** I CUI ELEMENTI SONO QUELLI  
DELL'INSIEME.

IN TAL MODO SI PUO' EVITARE CHE GLI ELEMENTI  
SIANO ASSOLUTAMENTE DEGLI INTERI, COME PURE  
DI SPRECARRE INUTILMENTE DELLO SPAZIO.

LE LISTE POSSONO ESSERE SIA **NON ORDINATE** CHE  
**ORDINATE** SULLA BASE DELLA RELAZIONE  
D'ORDINE ESISTENTE TRA GLI ELEMENTI  
DELL'INSIEME.



## REALIZZAZIONE CON LISTE NON ORDINATE

GLI ELEMENTI DELLA LISTA SONO QUELLI DELL'INSIEME. NEL CASO SI USINO REALIZZAZIONI CON STRUTTURE DINAMICHE, L'OCCUPAZIONE DI MEMORIA È PROPORZIONALE AL NUMERO DEGLI ELEMENTI PRESENTI NELL'INSIEME. UNA POSSIBILE DEFINIZIONE DI TIPO

posizione: tipo puntatore a cella

cella: tipo strutturato con componenti

- elemento di tipo `tipoelem`
- successivo di tipo `posizione`

insieme: alias per il tipo `posizione`

L'INSERIMENTO AVVIENE IN TESTA ALLA LISTA SEMPLICE CON CUI È REALIZZATO L'INSIEME



*FUNCTION APPARTIENE(i:TIPOELEM; A:INSIEME)→ BOOLEAN*

*VAR*

*P di tipo POSIZIONE, TROVATO di tipo BOOLEAN;*

*BEGIN*

*P← A*

*TROVATO← FALSE*

*WHILE (P ≠ NIL) AND (NOT TROVATO) DO*

*IF P.ELEMENTO=i THEN TROVATO←TRUE*

*ELSE P←P.SUCCESSIVO*

*APPARTIENE←TROVATO*

*END*

Realizzazione con variabile dinamica di tipo puntatore

```
PROCEDURE  INSERISCI (A:INSIEME; ELEM:TIPOELEM)
VAR
    B di tipo INSIEME; TEMP di tipo POSIZIONE;
BEGIN
    IF NOT APPARTIENE(ELEM,A)  THEN
        BEGIN
            TEMP←A
            B.ELEMENTO← ELEM
            B.SUCCESSIVO←TEMP
            A←B
        END
    END
END
```

```
PROCEDURE DIFFERENZA(A,B:INSIEME;C:INSIEME)
VAR
    P e TEMP di tipo POSIZIONE
BEGIN
    C←NIL
    P←A
    WHILE P ≠ NIL DO BEGIN
        IF NOT APPARTIENE(P.ELEMENTO, B) THEN
            BEGIN
                TEMP←C
                C.SUCCESSIVO←TEMP
                C.ELEMENTO← P.ELEMENTO
            END
        P←P.ELEMENTO
    END
END
```

## REALIZZAZIONE CON LISTE ORDINATE

SE È DEFINITA UNA RELAZIONE  $\leq$  DI ORDINAMENTO TOTALE SUGLI ELEMENTI L'INSIEME PUÒ ESSERE RAPPRESENTATO CON UNA LISTA ORDINATA PER VALORI CRESCENTI DEGLI ELEMENTI UTILIZZANDO DUE PUNTATORI CHE SCORRONO OGNUNO SU UN INSIEME.

LA REALIZZAZIONE DEGLI OPERATORI NON PRESENTA PARTICOLARI DIFFICOLTA'.

DI SEGUITO VIENE DATO COME ESEMPIO IL DIAGRAMMA DI FLUSSO DEL SOLO OPERATORE **INTERSEZIONE.**





PROCEDURE INTERSEZIONE (A,B:INSIEME; C:INSIEME)

VAR p,q,r di tipo POSIZIONE

BEGIN

C←NIL

p←A     {p punta al primo elemento della lista A}

q←B     {q punta al primo elemento della lista B}

r←C     {r punta al primo elemento della lista C, ora vuota}

WHILE (p ≠ NIL) AND (q ≠ NIL) DO

BEGIN

IF p.ELEMENTO= q.ELEMENTO THEN

   {elemento in A = elemento in B, quindi si inserisce in C}

   BEGIN

     r.ELEMENTO←p.ELEMENTO

     r← r.SUCCESSIVO

     p← p.SUCCESSIVO

     q←q.SUCCESSIVO

   END

ELSE

   {elementi in A e B diversi, si passa al successivo del minore senza inserire in C}

   IF p↑.ELEMENTO < q.ELEMENTO

     THEN p←p.SUCCESSIVO

     ELSE q←q.SUCCESSIVO

END

END;

# ESEMPI DI APPLICAZIONE DI INSIEMI

PROBLEMA: TROVARE I NUMERI PRIMI APPARTENENTI ALL'INTERVALLO  $2..n$  CON  $n > 2$

ALGORITMO (*SETACCIO DI ERATOSTENE*)

1. METTI TUTTI I NUMERI TRA 2 E  $n$  NEL “SETACCIO”
2. SCEGLI E RIMUOVI IL NUMERO IN “SETACCIO”
3. INCLUDI QUESTO NUMERO IN “NUMERI PRIMI”
4. RIMUOVI DAL “SETACCIO” TUTTI I MULTIPLI DI QUESTO NUMERO
5. SE IL “SETACCIO” NON È VUOTO RIPETI I PASSI 2- 5

SIA “SETACCIO” CHE “NUMERI PRIMI” SONO DEFINIBILI COME INSIEMI.



## IN PSEUDOCODICE

*PROSSIMO, J: integer*

*CREAINSIEME(SETAC)*

*CREAINSIEME(PRIMI)*

*FOR J=2 TO N DO INSERISCI(J,SETAC)*

*PROSSIMO  $\leftarrow$  2*

*REPEAT*

*WHILE NOT(APPARTIENE(PROSSIMO,SETAC))*

*DO PROSSIMO  $\leftarrow$  SUCC(PROSSIMO)*

*INSERISCI (PROSSIMO,PRIMI)*

*J  $\leftarrow$  PROSSIMO*

*WHILE J  $\leq$  N DO*

*CANCELLA(J,SETAC)*

*J  $\leftarrow$  J + PROSSIMO*

*UNTIL (INSIEME VUOTO(SETAC))*



CI SONO SPESSO DELLE APPLICAZIONI CHE NON RICHIEDONO L'USO DI TUTTE LE OPERAZIONI

**AD ESEMPIO:**

- TENERE LE REGISTRAZIONI DEGLI STIPENDI DEGLI IMPIEGATI DI UNA IMPRESA: NON INTERESSA ALCUNA STRUTTURA CHE RIGUARDA GLI IMPIEGATI E LE OPERAZIONI **UNIONE, INTERSEZIONE E DIFFERENZA** NON SONO NECESSARIE
- IMPLEMENTARE LA TABELLA DEI SIMBOLI: COME E' NOTO LA **TABELLA DEI SIMBOLI** DI UN COMPILATORE È USATA PER MEMORIZZARE I NOMI DELLE COSTANTI, DEI TIPI E DELLE VARIABILI DI UN PROGRAMMA SORGENTE.

TRADIZIONALMENTE SI USANO STRUTTURE PARTICOLARI **(INSIEMI DINAMICI, DIZIONARI)** CHE AVREMO MODO DI VEDERE IN SEGUITO



# L'INSIEME DINAMICO

## Struttura dati “generale”: *insieme dinamico*

- ❑ Può crescere, contrarsi, cambiare contenuto
- ❑ Operazioni base: inserimento, cancellazione, verifica di appartenenza/contenimento
- ❑ Il tipo di insieme (= struttura) dipende dalle operazioni

## Elementi

- ❑ Elemento: oggetto *indirizzabile*
- ❑ Composto da:
  - ❑ campo chiave di identificazione
  - ❑ dati
  - ❑ campi che fanno riferimento ad altri elementi dell'insieme



## UN ALTRO TIPO DI INSIEME: il Dizionario

**Il dizionario rappresenta il concetto matematico di relazione univoca**

- ☐ Relazione  $R : D \rightarrow C$
- ☐ Insieme  $D$  è il dominio (elementi detti **chiavi**)
- ☐ Insieme  $C$  è il codominio (elementi detti **valori**)
- ☐ Associazione **chiave-valore**

**Operazioni ammesse:**

- ☐ ottenere il valore associato ad una particolare chiave (se presente), o nessun valore
- ☐ inserire una nuova associazione chiave- valore, cancellando eventuali associazioni precedenti;
- ☐ rimuovere un'associazione chiave-valore esistente



# Struttura dati per insiemi disgiunti

## Motivazioni

In alcune applicazioni siamo interessati a gestire *insiemi disgiunti* di oggetti

Esempio: componenti di una gerarchia o di una tassonomia

❑ Operazioni fondamentali:

❑ unire più insiemi

❑ identificare l'insieme a cui appartiene un oggetto

## Struttura dati

❑ Una collezione  $S = \{ S_1, S_2, \dots, S_n \}$  di insiemi dinamici disgiunti

❑ Ogni insieme è identificato da un *rappresentante* univoco



# Scelta del rappresentante

Il rappresentante può essere un qualsiasi membro dell'insieme  $S_i$

- ☐ Operazioni di ricerca del rappresentante su uno stesso insieme devono restituire sempre lo stesso oggetto
- ☐ Solo in caso di unione con altro insieme il rappresentante può cambiare

Il rappresentante può essere un elemento specifico dell'insieme

- ☐ Si devono definire le caratteristiche degli insiemi e una regola per caratterizzare il rappresentante
- ☐ Esempio: l'elemento più piccolo/grande di un insieme





# Insiemi disgiunti

**QUEST'ULTIMO TIPO DI INSIEME E' NOTO COME  
MFSET(MERGE-FIND-SET)**

**UN MFSET È UNA *PARTIZIONE DI UN INSIEME FINITO IN SOTTOINSIEMI DISGIUNTI DETTI COMPONENTI.***

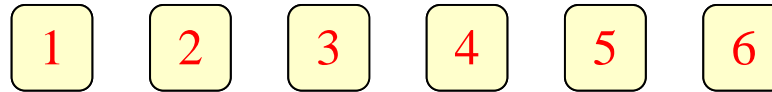
**LE OPERAZIONI CONSENTITE PERMETTONO DI :**

- **STABILIRE A QUALE COMPONENTE APPARTIENE UN ELEMENTO GENERICO**
- **UNIRE DUE COMPONENTI DISTINTE IN UNA SOLA COMPONENTE DISTRUGGENDO LE VECCHIE COMPONENTI MA LASCIANDO INALTERATE LE COMPONENTI RIMANENTI**

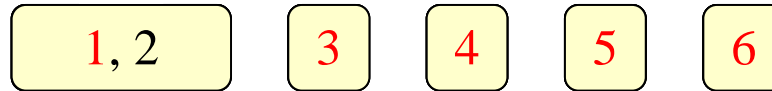


# Esempio

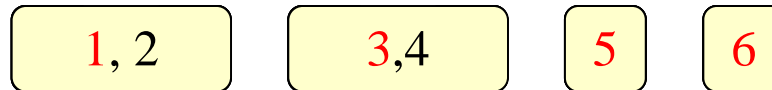
Mfset di 6 elementi



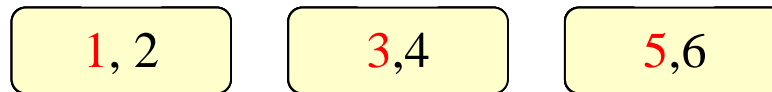
Fondi 1 e 2



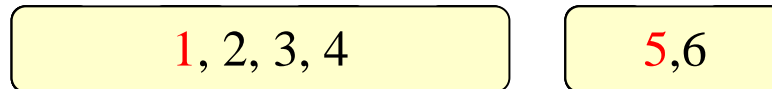
Fondi 3 e 4



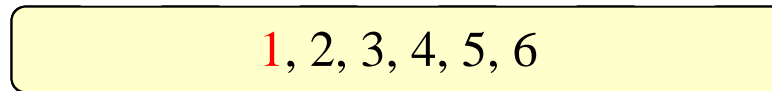
Fondi 5 e 6



Fondi 1 e 3



Fondi 1 e 5



# ALGEBRA DI MFSET (MERGE-FIND-SET)

## SPECIFICA

### TIPI :

**INSIEME** (FAMIGLIA DI INSIEMI COSTITUITA DA ELEMENTI DI TIPO **TIPOELEM**),

**MFSET** (FAMIGLIA DI PARTIZIONI DI INSIEMI DI ELEMENTI DI TIPO **TIPOELEM**)

**COMPONENTE** (SOTTOINSIEME DI INSIEME, CHE E' ELEMENTO DI **MFSET**)

**BOOLEAN**

### OPERATORI :

**CREAMFSET:** (INSIEME)  $\rightarrow$  MFSET

**CREAMFSET (A)= S**

POST : S È UNA FAMIGLIA DI  $n=|A|$  COMPONENTI  $C_1, C_2, \dots, C_n$  OGNUNO DELLE QUALI CONTIENE UNO E UN SOLO ELEMENTO DI A E TALI CHE  $\bigcup C_i = A \quad 1 \leq i \leq n$



**FONDI** : (TIPOELEM, TIPOELEM, MFSET)  $\rightarrow$  MFSET

**FONDI** (X, Y, S) = S'

PRE : X E Y SONO APPARTENENTI A COMPONENTI  
DISTINTE DI S

POST : S' È DA TUTTE LE COMPONENTI CHE NON  
CONTENGONO X E Y E DA UNA NUOVA

**TROVA** : (TIPOELEM, MFSET)  $\rightarrow$  **COMPONENTE**

**TROVA** (X, S) = C

PRE : X APPARTIENE AD UNA COMPONENTE DI S

POST : C È L'IDENTIFICATORE DELLE  
COMPONENTE CUI X APPARTIENE



IN LETTERATURA SI PUÒ TROVARE ANCHE UN ALTRO OPERATORE **TROVA** TESO A VERIFICARE SE DATI DUE ELEMENTI QUESTI APPARTENGONO ALLA STESSA COMPONENTE

IN QUESTO CASO:

**TROVA**: (TIPOELEM , TIPOELEM , MFSET )  $\rightarrow$  BOOLEAN

**TROVA** (X , Y , S)= b

PRE : X E Y APPARTENGONO A COMPONENTI DI S

POST : b= VERO SE X E Y  $\in$  ALLA STESSA COMPONENTE

b= FALSO

ALTRIMENTI



UNA REALIZZAZIONE SEMPLICE DI MFSET CONSISTE NEL CREARE UNA LISTA PER OGNUNO DEGLI INSIEMI. L'ELEMENTO IN TESTA E' IL RAPPRESENTANTE. LA OPERAZIONE **CREAMFSET** CREA N LISTE UNA PER OGNI ELEMENTO

SIA  $A = \{1, 2, 3, 4, 5, 6, 7\}$

**CREAMFSET** RESTITUISCE

$S = \{ [1], [2], [3], [4], [5], [6], [7] \}$

POICHÉ

$TROVA(1, S) \neq TROVA(5, S)$  oppure

**$TROVA(1, 5, S) = \text{False}$**

SI PUO' APPLICARE LA OPERAZIONE DI FUSIONE

$FONDI(1, 5, S) = S'$

$S' = \{ [2], [3], [4], [1, 5], [6], [7] \}$

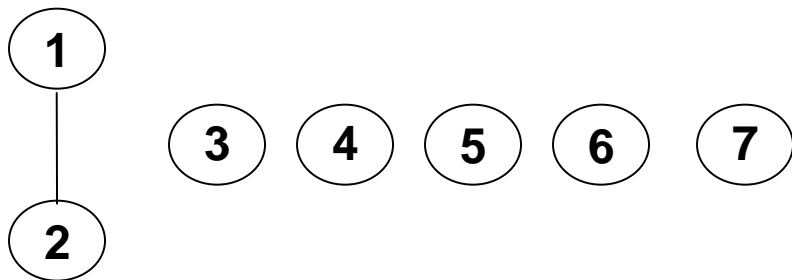


**REALIZZAZIONI PIU' EFFICACI DI MFSET  
PREVEDONO L'USO DI STRUTTURE AD ALBERO, IN  
PARTICOLARE TRAMITE FORESTE DI ALBERI  
RADICATI CHE SARANNO PRESENTATE PIU' TARDI.**

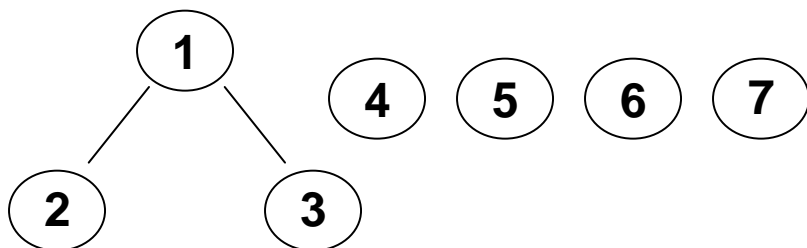




**FONDI (1,2,S)**



**FONDI (1,3,S)**



**FONDI (5,6,S)**

