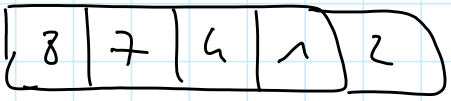
mercoledì 4 ottobre 2017 16:12

AVVISO

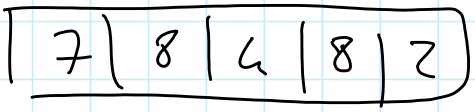
LA LEZIONE DI VENERDI 6/10

É ANNULLATA

→ VERIFICARE CHE TUTTI GLI ELEMENTI DI UN ARRAY SIANO DIVERSI TRA LORO

a 

✓

b 

✗

DEVO CONFRONTARE OGNI ELEMENTO CON TUTTI GLI ALTRI

FORMALIZZIAMO IL PROBLEMA

$$\forall i \in [0, dim) . \forall j \in [0, dim) . (a[i] != a[j] \vee i == j)$$

FISSIAMO i E RISOLVIAMO QUESTO SOTTOPROBLEMA

$$\exists j \in [0, dim) . (a[i] == a[j] \wedge i != j)$$

FALSA

VADO A VEDERE SE CI SONO DUE ELEMENTI CON INDICI DIVERSI MA I CUI VALORI NELL'ARRAY SONO UGUALI

```
int diversoi (int a[], int dim, int i) {
    int j=0;
    int ok=1;
    while (j<dim && ok) {
        if (a[i]==a[j] && i!=j)
            ok=0;
        j++;
    }
    return ok;
}
```

ADESSO CHE HO DEFINITO diversoi

LA FORMULA

$$\forall i \in [0, dim) . \forall j \in [0, dim) . (a[i] != a[j] \vee i == j)$$

DIVENTA EQUIVALENTE

$$\forall i \in [0, dim) . diversoi(a, dim, i)$$

CIOÉ DEVO VERIFICARE

$$\exists i \in [0, dim) . (!\, diversoi(a, dim, i))$$

È FALSO

```
int TuttiDiversi (int a[], int dim) {
    int i=0;
    int ok=1;
    while (i<dim && ok) {
      if (!diversoi(a, dim, i))
            ok=0;
    } i++;
    return ok;
}
```

PER OGNI ELEMENTO DELL'ARRAY VADO A VERIFICARE CHE SIA DIVERSO DA TUTTI GLI ALTRI

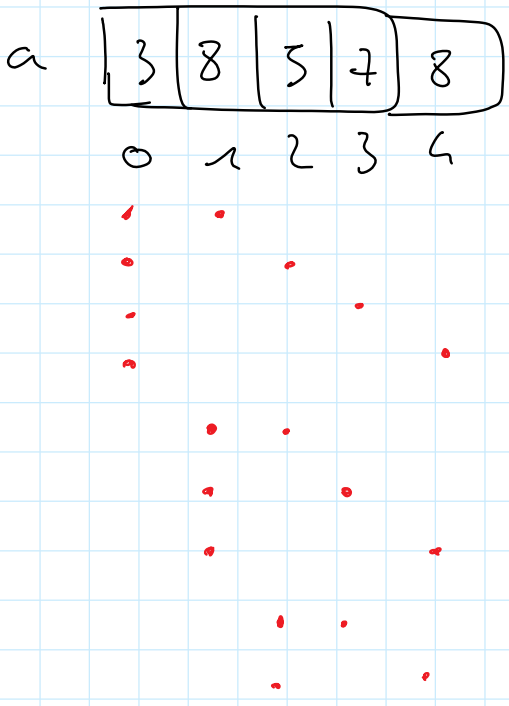
a

8	8	2	7
0	1	2	3

NELLA SOLUZIONE CHE ABBIAMO VISTO OGNI COPPIA DI ELEMENTI VIENE CONFRONTATA DUE VOLTE ....

ES. $i = 2$ $j = 3$

E $i = 3$ $j = 2$



È SUFFICIENTE CONFRONTARE OGNI ELEMENTO CON QUELLI CHE SEGUONO.

CON QUELLI CHE LO PRECEDONO L'HO GIÀ CONFRONTATO.

RIFORMULIAMO IL PROBLEMA PER RENDERE LA SOLUZIONE PIÙ EFFICIENTE (EVITIAMO CONFRONTI INUTILI)

INVECE DI CALCOLARE

$$\forall i \in [0, dim) \,.\, \forall j \in (0, dim) \,.\, (a[i] != a[j] \;\vee\; i == j)$$

CALCOLIAMO

$$[\quad \forall i \in [0, dim-1) \;\; \forall j \in [i+1, dim) \,.\, a[i] != a[j]$$

- $dim-1$: L'ULTIMO ELEMENTO NON HA SUCCESSORI
- $[i+1, dim)$: CONSIDERO SOLO I SUCCESSORI DI i
- $i \in j$ SONO SICURAMENTE DIVERSI ($j > i$)

$$\forall i \in [0, dim-1). \forall j \in [i+1, dim) . \; a[i] != a[j]$$

$$\exists j \in [i+1, dim) . \; a[i] == a[j] \quad \text{FALSO}$$

NOTA: SE CHIAMO $a[i] = x$ QUESTO È QUASI UGUALE ALLA MEMBER SOLO CHE PARTO DALLA POSIZIONE i+1

```
int membri (int a[], int dim, int i) {
    int j = i+1;
    int ok = 1;
    while (j < dim && ok) {
        if (a[i] == a[j])
            ok = 0;
        j++;
    }
    return ok;
}
```

ORA DEVO RISOLVERE

$$\forall i \in [0, dim) . !membri(a, dim, i)$$

Quindi

```
int TuttiDivisi (int a[], int dim) {
    int i = 0;
    int ok = 1;
    while (i < dim && ok) {
        if ( member i (a, dim, i))
            ok = 0;
        i++;
    }
    return ok;
}
```

ALTRA SOLUZIONE NON MODULARE

(OK SE IL PROBLEMA È "SEMPLICE"
ALTRIMENTI È MEGLIO MODULARIZZARE,
OSSIA DEFINIRE PIÙ FUNZIONI CHE
RISOLVONO I SOTTO PROBLEMI)

→ USO CICLI ANNIDATI (DUE CICLI UNO DENTRO L'ALTRO)

```
int TuttiDiversi (int a[], int dim) {
    int i=0;
    int ok=1;
    while (i<dim-1 && ok) {
        int j=i+1;
        while (j<dim && ok) {
            if (a[i]==a[j])
                ok=0;
            j++;
        }
        i++;
    }
    return ok;
}
```

STESSA VARIABILE OK

a | 8 | 7 | 4 | 7 |

0 ↗ 2

a | 8 | 7 | 4 | 7 |

0 1 2 3

i i i i

a[i] a[j] a[j] a[j]

i i i

a[i] a[j] a[j]

10 4 4

j 1 2 3 4 3

ok 1 0

→ PROBLEMI CON 2 ARRAY

ESEMPI:

→ dati due array a e b di dimensioni dima e dimb verificare se TUTTI gli elementi del primo sono anche elementi del secondo:

→ POSSONO AVERE DIMENSIONI DIVERSE

$$\forall i \in [0, dima) . \exists j \in [0, dimb) . a[i] == b[j]$$

TROVATE LA SOLUZIONE NEGLI APPUNTI DEL CORSO A (BARBUTI) DEL 2/10

→ dati due array a e b di dimensione dima e dimb verificare se esiste un elemento in a che corrisponde alla somma degli elementi di b

$$\exists i \in [0, dima) . a[i] = \sum_{j \in [0, dimb)} b[j]$$

$$\exists\, i \in [0, dima) \,.\, a[i] = \sum_{j \in [0, dimb)} b[j]$$

```
int somma (int b[], int dimb) {
    int s = 0
    for (int i=0; i<dimb; i++)
        s += b[i];
    return s;
}
```

ACCUMULAZIONE

PRENDE ENTRAMBI GLI ARRAY

VERIFICA

```
int check (int a[], int dima, int b[], int dimb)
    int i=0;
    int Trovato=0;
    while (i<dima && !Trovato) {
        if (a[i] == somma(b, dimb))
            Trovato = 1;
        i++
    }
    return Trovato;
}
```

PER RENDERE IL PROGRAMMA PIU' EFFICIENTE POTREI CHIAMARE SOMMA UNA SOLA VOLTA PRIMA DEL WHILE ASSEGNANDO IL RISULTATO A UNA VARIABILE

ESERCIZI:

- SCRIVERE UNA FUNZIONE CHE, DATO UN ARRAY a DI DIMENSIONE dim, VERIFICA LA SEGUENTE PROPRIETÀ:

$$\forall i \in [0, dim-1) \,.\, a[i] < a[i+1]$$

- SCRIVERE UNA FUNZIONE CHE, DATO UN ARRAY a DI DIMENSIONE dim, VERIFICA LA SEGUENTE PROPRIETÀ:

$$\exists i \in [0, dim) \,.\, a[i] = \sum_{j \in [0,i)} a[j]$$

- SCRIVERE UNA FUNZIONE CHE, DATO UN ARRAY a, DI DIMENSIONE dim, VERIFICA LA SEGUENTE PROPRIETÀ:

$$\sum_{\substack{i \in [0,dim) \\ a[i]>0}} a[i] == -\sum_{\substack{i \in [0,dim) \\ a[i]<0}} a[i]$$

ANCORA ESERCIZI

→ DATO UN ARRAY a DI DIMENSIONE dim

CALCOLARE

$$\#\left\{ i \mid i \in [1, dim-1) \wedge a[i] == \sum_{j \in [0,i)} a[j] \wedge a[i] == \sum_{j \in [i+1, dim)} a[j] \right\}$$

→ DATI DUE ARRAY a E b DI DIMENSIONI dima E dimb VERIFICARE LA SEGUENTE PROPRIETA':

$$\forall i \in [0, dima).\ \exists j \in [0, dimb).\ \exists k \in [0, dimb).\ (a[i] == b[j] \wedge a[i] == b[k])$$