

6 Condizioni

6.1 Condizioni su array

Dato un array **a** di dimensione N , voglio verificare se la proprietà P vale per tutti gli elementi dell'array.

$$\forall i \in [0, N). \mathcal{P}(a[i]) \quad (4)$$

Esempio 6.1. Verifico che tutti gli elementi dell'array siano dispari.

$$\forall i \in [0, N). a[i] \quad (5)$$

```

1  int check_array_dispari(int a[], size_t dim) {
2      int indice = 0;
3      while (indice < dim && a[indice]%2 == 1){
4          indice++;
5      }
6      if (indice == dim) {
7          return 1;
8      } else {
9          return 0;
10     }
11 }
```

Listing 8: Verifica di proprietà su tutti gli elementi mathescape

Blocco lo scorrimento dell'array quando la proprietà **NON** viene soddisfatta almeno una volta.

Se invece voglio verificare che la proprietà P valga per almeno un elemento:

$$\exists i \in [0, N). \mathcal{P}(a[i]) \quad (6)$$

Esempio 6.2. Verifico che almeno un elemento dell'array è uguale a 26.

$$\exists i \in [0, N). a[i] == 26 \quad (7)$$

```

1  int esiste_in_array(int a[], size_t dim, in n) {
2      size_t indice = 0;
3      _Bool trovato = 0;
4      while (indice < dim && !trovato){
5          if(a[indice] == n) {
6              trovato = 1;
7          }
8          indice++;
9      }
10     return trovato;
11 }
```

Listing 9: Verifica di proprietà su almeno un elemento

Blocco lo scorrimento dell'array nel momento in cui trovo un elemento che soddisfa la proprietà, utilizzando un *flag*.

6.2 Condizioni su matrici

Una **matrice** è un array di array. Può essere *multidimensionale* $N \times M$ e voglio verificare se tutti i suoi elementi oppure solo uno di essi verificano una proprietà P .

$$\forall i \in [0, N), \forall j \in [0, M). \mathcal{P}(a[i, j]) \quad (8)$$

$$\exists i \in [0, N), \exists j \in [0, M). \mathcal{P}(a[i, j]) \quad (9)$$

Definizione 6.1 (Matrice quadrata). Una matrice è **quadrata** se a lo stesso numero di righe e di colonne. In questo caso per scorrerla si può usare un solo indice:

$$\exists i, j \in [0, N). \mathcal{P}(a[i, j]) \quad (10)$$

Esempio 6.3. Verifico se tutti gli elementi della matrice sono positivi.

$$\forall i \in [0, N), \forall j \in [0, M). a[i, j] > 0 \quad (11)$$

```

1  int check_matrice_pos(int a[][COL], size_t dim) {
2      size_t row, col;
3      row = col = 0;
4      while (row < dim && a[row][col] > 0) {
5          col = 0;
6          while (col < COL && a[row][col] > 0) {
7              col++;
8          }
9          if (col == COL) {
10             row++;
11         }
12     }
13     if (row == dim && col == COL) {
14         return 1;
15     }
16     else {
17         return 0;
18     }
19 }
```

Listing 10: Verifica di proprietà su tutti gli elementi della matrice

Definizione 6.2 (Matrice simmetrica). *Una matrice è **simmetrica** se è quadrata e se le posizioni simmetriche rispetto alla diagonale principale contengono gli stessi elementi.*

Definizione 6.3 (Matrice triangolare). *Una matrice è **triangolare** superiore o inferiore se le posizioni rispettivamente sopra o sotto la diagonale contengono tutti 0.*

Definizione 6.4 (Matrice tridiagonale). *Una matrice **tridiagonale** può avere elementi non nulli solo sulla diagonale principale e la sua diagonale superiore ed inferiore.*

6.3 Contare elementi che verificano una proprietà

Dato un array **a** di dimensione N per contare tutti gli elementi che verificano una proprietà P :

$$\#\{i | i \in [0, N - 1] \wedge \mathcal{P}(a[i])\} \quad (12)$$

Data invece una matrice **a** di dimensione $N \times M$:

$$\#\{(i, j) | i \in [0, N - 1] \wedge j \in [0, M - 1]\} \quad (13)$$