

Informatica 1

Corso di Laurea Triennale in Matematica

Gianluca Rossi

`gianluca.rossi@uniroma2.it`

Dipartimento di Matematica
Università di Roma "Tor Vergata"

8: Array Multidimensionali



Array multidimensionali

tipo $d2[N][M]$

Crea un array di dimensione **2** (matrice) di **N** righe e **M** colonne. Ogni elemento dell'array è di tipo *tipo*.

$d2[i][j]$

con $0 \leq i < N$ e $0 \leq j < M$ si riferisce all'elemento della $(i + 1)$ -esima riga $(j + 1)$ -esima colonna della matrice $d2$.

La dimensione dell'array può essere anche 3, 4, ...

tipo $d3[N][M][H]$



Esempio: Prodotto matrice \times vettore

$$M \in \mathbb{R}^{n \times n}$$

$$v \in \mathbb{R}^n$$



$$p = M \times v \in \mathbb{R}^n$$

$$p_i = \sum_{j=0}^{n-1} M_{i,j} \cdot v_j$$

```
#define n 3
```

```
main(){
```

```
    int M[n][n] = {{1,3,-2}, {2,-4,5}, {-1,3,9}};
```

```
    int v[n] = {7,1,-2};
```

```
    int p[n] = {0};
```

```
    int i,j;
```

```
    for(i = 0; i < n; i++)
```

```
        for(j = 0; j < n; j++)
```

```
            p[i] = p[i] + M[i][j] * v[j];
```

```
}
```



```
short a[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

- Un vettore di **2** elementi ognuno dei quali è un vettore di **3** elementi.
- `a[0]` o `a` è il puntatore al primo vettore di **3** elementi;
- `a[1]` o `a+1` è il puntatore al secondo vettore di **3** elementi;
- `*(a+1)` è il secondo vettore di **3** elementi, ovvero il puntatore al suo primo elemento;

<code>a[0][0] = 1</code>	<i>F18</i>
	<i>F19</i>
<code>a[0][1] = 2</code>	<i>F1A</i>
	<i>F1B</i>
<code>a[0][2] = 3</code>	<i>F1C</i>
	<i>F1D</i>
<code>a[1][0] = 4</code>	<i>F1E</i>
	<i>F1F</i>
<code>a[1][1] = 5</code>	<i>F20</i>
	<i>F21</i>
<code>a[1][2] = 6</code>	<i>F22</i>
	<i>F23</i>



Memorizzazione

$a[0][0]$	1	$F18$ $F19$
$a[0][1]$	2	$F1A$ $F1B$
$a[0][2]$	3	$F1C$ $F1D$
$a[1][0]$	4	$F1E$ $F1F$
$a[1][1]$	5	$F20$ $F21$
$a[1][2]$	6	$F22$ $F23$
...		...
$*a \circ a[0]$	$F18$	$FB7$ $FB8$ $FB9$ FBA
$*(a+1) \circ a[1]$	$F1E$	$FB B$ FBC

- $*a = F18$
- $*(a+1) = F1E$
- $*(a+1) + 1 = F20$
- $*(*(a+1)) = 4$
- $*(*(a+1) + 2) = 6$
- $(*a + 4) = F20$
- $*(*a + 4) = 5$



Ottimizzazioni

```
#include<stdio.h>
#define N 1000
#define M 1500
main(){
    int a[N][M], r, c, i;
    for(i = 0, r = 0; r < N; r++)
        for(c = 0; c < M; c++, i++)
            a[r][c] = i; /* *(a+r) + c) */
}
```

```
#include<stdio.h>
#define N 1000
#define M 1500
main(){
    int a[N][M], i;
    int *p, *lp;
    lp = *(a+(N-1)) + M;
    for(p = *a, i = 0; p < lp; p++, i++)
        *p = i;
}
```

Contiamo *solo* le operazioni eseguite $N \cdot M$ volte.

- Test + 2 Incrementi (**for**);
- 2 Somme + 2 Moltiplicazioni + Assegnazione ($a[r][c] = i$)

$$8 \cdot N \cdot M$$

- Test + 2 Incrementi (**for**);
- Assegnazione ($*p = i$)

$$4 \cdot N \cdot M$$

