

Διάλεξη 10η: Πολυδιάστατοι Πίνακες

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Εισαγωγή στην Επιστήμη Υπολογιστών

Βασίζεται σε διαφάνειες του Κ. Παναγιωτάκη



Πίνακες πολλών διαστάσεων

- Πίνακες πολλών διαστάσεων

- Πίνακες με γραμμές και στήλες ($m \times n$ πίνακας)
- Όπως και στα μαθηματικά: πρώτα γραμμή και μετά στήλη

Δισδιάστατος πίνακας

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]



Πίνακες πολλών διαστάσεων (2)

- Δήλωση πολυδιάστατου πίνακα

- `int c[2][4];`

Γενική μορφή

```
type array_name[rows][columns];
```

- Χρήση πολυδιάστατου πίνακα

- `c[i][j] = 3;`
- `int *ptr = c[i];`

Γενική μορφή

```
array_name[row][column]
```



Πίνακες πολλών διαστάσεων (3)

- Αποθήκευση στη μνήμη

- Πρώτα η σειρά 0, μετά η σειρά 1, κ.ο.κ.

c[0][0]	10
c[0][1]	22
c[0][2]	22
c[0][3]	30
c[1][0]	-11
c[1][1]	1
c[1][2]	14
c[1][3]	32

- Αρχικοποίηση: όπως η αρχικοποίηση πίνακα, αλλά με πίνακες ως στοιχεία

- `int b[2][2] = { { 1, 2 }, { 3, 4 } };`
 - Τα εσωτερικά `{ }` διαχωρίζουν γραμμές του πίνακα
 - Αν δεν είναι αρκετά τα στοιχεία, τα υπόλοιπα γίνονται 0
 - `int b[2][2] = { { 1 }, { 3, 4 } };`



Πολυδιάστατοι Πίνακες ως Ορίσματα

- Πάντα πρέπει να δηλώνεται το μέγεθος κάθε διάστασης, εκτός από την πρώτη διάσταση (αριθμός γραμμών)

Παράδειγμα

```
int f(int a[][10][20])
{
    ...
}
int main(void)
{
    int b[40][10][20];
    f(b);
}
```

- Γιατί μας υποχρεώνει η γλώσσα σε αυτόν τον περιορισμό;



Παράδειγμα

students.c

```
#include <stdio.h>
#include <string.h>
#define STUDENTS 3
#define EXAMS 4

/* function declarations */
int min(const int[][EXAMS], int, int);
int max(const int[][EXAMS], int, int);
double average(const int[], int);
void print_array(const int[][EXAMS],
                 int, int);

/* function definitions */
double average(const int student_grades[],
               int tests) {
    int i, total = 0;
    for(i = 0; i <= tests - 1; i++) {
        total = total + student_grades[i];
    }
    return (double) total / tests;
}

int main(void) {
    int s;
    const int grades[STUDENTS][EXAMS] =
        { { 77, 68, 86, 73 },
          { 96, 87, 89, 78 },
          { 70, 90, 86, 81 } };

    printf("The array is:\n");
    print_array(grades, STUDENTS, EXAMS);
    printf("\n\n"
           "Lowest grade: %d\n"
           "Highest grade: %d\n",
           min(grades, STUDENTS, EXAMS),
           max(grades, STUDENTS, EXAMS));

    for(s = 0; s <= STUDENTS - 1; s++) {
        printf("Average for student %d: %.2f\n",
              s, average(grades[s], EXAMS));
    }
    return 0;
}
```

Παράδειγμα (2)

students.c

```
int min(const int grades[][EXAMS],  
        int students, int tests)  
{  
    int i, j, lowGrade = 100;  
    for(i = 0; i <= students -1; i++) {  
        for(j = 0; j <= tests -1; j++) {  
            if(grades[i][j] < low_grade) {  
                low_grade = grades[i][j];  
            }  
        }  
    }  
    return low_grade;  
}
```

```
int max(const int grades[][EXAMS],  
        int students, int tests)  
{  
    int i, j, high_grade = 0;  
    for(i = 0; i <= students -1; i++) {  
        for(j = 0; j <= tests -1; j++) {  
            if(grades[i][j] > high_grade) {  
                high_grade = grades[i][j];  
            }  
        }  
    }  
    return high_grade;  
}
```



Παράδειγμα (3)

students.c

```
void print_array(const int grades[][EXAMS],
                int students, int tests)
{
    int i, j;
    char array_name[] = "student_grades";

    for(i = 0; i <= strlen(array_name) + 3; i++) {
        printf(" ");
    }
    for(i = 0; i <= tests -1; i++) {
        printf(" [%d]", i);
    }
    for(i = 0; i <= students -1; i++) {
        printf("\nstudent_grades[%d]", i);
        for(j = 0; j <= tests -1; j++) {
            printf("%5d", grades[i][j]);
        }
    }
}
```


Παράδειγμα (3)

Έξοδος

The array is:

	[0]	[1]	[2]	[3]
student_grades[0]	77	68	86	73
student_grades[1]	96	87	89	78
student_grades[2]	70	90	86	81

Lowest grade: 68

Highest grade: 96

The average grade for student 0 is 76.00

The average grade for student 1 is 87.50

The average grade for student 2 is 81.75



Υπολογισμός Mean, Median και Mode

- Mean – Μέσος όρος
 - 1, 1, 2, 2, 40
 - $\text{Mean} = (1+1+2+2+40) / 5 = 9.2$
- Median – Ο μεσαίος αριθμός
 - 1, 2, 3, 4, 40
 - Το median είναι 3
- Mode – Ο πιο συχνός αριθμός
 - 1, 1, 1, 2, 3, 3, 4, 40
 - Το mode είναι 1



data-analysis.c

```
#include <stdio.h>
#define SIZE 99
/* function declarations */
void mean(const int []);
void median(int []);
void mode(int [], const int []);
void bubble_sort(int []);
void print_array(const int []);

/* function definitions */
void mean(const int answer[]) {
    int i, total = 0;
    printf("***** Mean *****\n");
    for(i = 0; i <= SIZE -1; i++) {
        total += answer[i];
    }
    printf("The mean is the average value of the data items.\n"
        "The mean is equal to the total of all the data\n"
        "items divided by the number of data items ( %d ).\n"
        "The mean value for this run is: %d / %d = %.4f\n\n",
        SIZE, total, SIZE, (double) total / SIZE );
```

Παράδειγμα (2)

data-analysis.c

```
void median(int answer[]) {  
    printf("***** Median *****\n"  
        "The unsorted array of responses is:");  
    print_array(answer);  
    bubble_sort(answer);  
    printf("\n\nThe median is element %d of\n"  
        "the sorted %d element array.\n"  
        "For this run the median is %d\n\n",  
        SIZE / 2, SIZE, answer[ SIZE / 2 ] );  
}
```

```
void bubble_sort(int a[]) {  
    int pass, j, hold;  
    for(pass = 1; pass <= SIZE - 1; pass++) {  
        for(j = 0; j <= SIZE - 2; j++) {  
            if(a[j] > a[j+1]) {  
                hold = a[j];  
                a[j] = a[j+1];  
                a[j+1] = hold;  
            }  
        }  
    }  
}
```

Παράδειγμα (3)

data-analysis.c

```
void mode(int freq[], const int answer[]) {  
    int rating, j, h, largest = 0, mode_value = 0;  
  
    printf("***** Mode *****\n");  
    for(rating = 1; rating <= 9; rating++) {  
        freq[rating] = 0;  
    }  
    for(j = 0; j <= SIZE -1; j++) {  
        ++freq[answer[j]];  
    }  
    printf("%s%11s%19s\n", "Response", "Frequency", "Histogram");  
    for(rating = 1; rating <= 9; rating++) {  
        printf(" %8d%11d", rating, freq[rating]);  
        if(freq[rating] > largest) {  
            largest = freq[rating];  
            mode_value = rating;  
        }  
        for(h = 1; h <= freq[rating]; h++) {  
            printf("*");  
        }  
        printf("\n");  
    }
```

Παράδειγμα (4)

data-analysis.c

```
void print_array(const int a[]) {  
    int j;  
    for(j = 0; j <= SIZE -1; j++) {  
        if(j % 20 == 0) {  
            printf("\n");  
        }  
        printf(" %2d", a[j]);  
    }  
}  
  
int main(void) {  
    int frequency[10] = {0};  
    int response[SIZE] = {  
        6, 7, 8, 9, 8, 7, 8, 9, 8, 9, 7, 8, 9, 5, 9, 8, 7, 8, 7, 8, 6, 7, 8, 9, 3,  
        9, 8, 7, 8, 7, 7, 8, 9, 8, 9, 8, 9, 7, 8, 9, 6, 7, 8, 7, 8, 7, 9, 8, 9, 2,  
        7, 8, 9, 8, 9, 8, 9, 7, 5, 3, 5, 6, 7, 2, 5, 3, 9, 4, 6, 4, 7, 8, 9, 6, 8,  
        7, 8, 9, 7, 8, 7, 4, 4, 2, 5, 3, 8, 7, 5, 6, 4, 5, 6, 1, 6, 5, 7, 8, 7 };  
    mean(response);  
    median(response);  
    mode(frequency, response);  
    return 0;  
}
```

Παράδειγμα: Έξοδος

***** Mean *****

The mean is the average value of the data items.
The mean is equal to the total of all the data items divided by the number of data items (99).
The mean value for this run is: $681 / 99 = 6.8788$

***** Median *****

The unsorted array of responses is:

```
6 7 8 9 8 7 8 9 8 9 7 8 9 5 9 8 7 8 7 8
6 7 8 9 3 9 8 7 8 7 7 8 9 8 9 8 9 7 8 9
6 7 8 7 8 7 9 8 9 2 7 8 9 8 9 8 9 7 5 3
5 6 7 2 5 3 9 4 6 4 7 8 9 6 8 7 8 9 7 8
7 4 4 2 5 3 8 7 5 6 4 5 6 1 6 5 7 8 7
```

The median is element 49 of
the sorted 99 element array.
For this run the median is 7

***** Mode *****

Response	Frequency	Histogram
1	1	*
2	3	***
3	4	****
4	5	*****
5	8	*****
6	9	*****
7	23	*****
8	27	*****
9	19	*****

The mode is the most frequent value.
For this run the mode is 8 which occurred 27 times.

