Multidimensional arrays

Goldsmiths Computing

Motivation

Sometimes the data that you want to store is naturally expressed as a table with more than one dimension.

Definition

A multidimensional array is an array that is subscipted using more than one index

NB: the “multidimensional” in multidimensional arrays refers to the subscripting, not the data that is stored:

linear array of 3-component colours Vector

linear array of 3-dimensional vectors Vector

2d array of grayscale values Multidimensional array

3d array of temperature values Multidimensional array

Operations

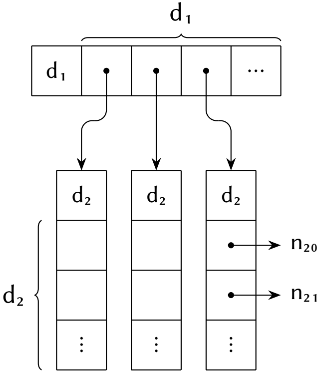
size
: return the number of elements in the multidimensional array

select[k,m,...,n]
: return the element at position k in the first dimension, m in the second, ..., and n in the last dimension

store![o,k,m,...,n]
: set the element at position k in the first dimension, m in the second, ..., and n in the last dimension to o.

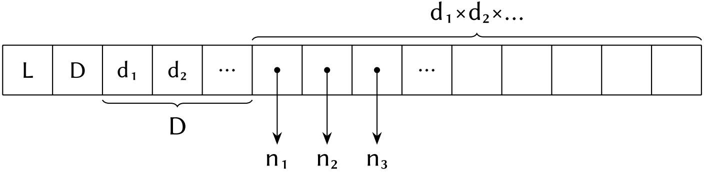
Implementation: Iliffe vector

Array of references to lower-dimensional arrays:



Implementation: dope vector

One-dimensional array with extra metadata (the “dope” on the array):



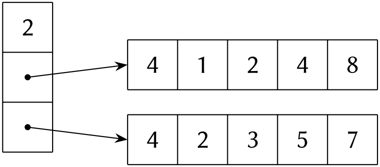
Implementation: example

Storing the 2×4 matrix

$$\begin{pmatrix} 1 & 2 & 4 & 8 \\ 2 & 3 & 5 & 7 \end{pmatrix}$$

Iliffe vector

Row-major ordering (compare earlier Iliffe vector diagram)



Implementation: example

Storing the 2×4 matrix

$$\begin{pmatrix} 1 & 2 & 4 & 8 \\ 2 & 3 & 5 & 7 \end{pmatrix}$$

dope vector

Row-major ordering

11	2	2	4	1	2	4	8	2	3	5	7

Implementation: example

Storing the 2×4 matrix

$$\begin{pmatrix} 1 & 2 & 4 & 8 \\ 2 & 3 & 5 & 7 \end{pmatrix}$$

dope vector

Column-major ordering

11	2	2	4	1	2	2	3	4	5	8	7

Size

Iliffe vector

Require: A :: two-dimensional (Illife) array
  function SIZE(A)
    return LENGTH(A) × LENGTH(A[0])
  end function

dope vector

Require: A :: multidimensional (dope) array
  function SIZE(A)
    D ⟵ A[0]
    result ⟵ 1
    for $0 \le d < D$ **do**
      result ⟵ result × A[1+d]
    end for
    return result
  end function

Select

Iliffe vector

Require: A :: multidimensional (Iliffe) array
Require: ks :: list of indices

```
function SELECT(A,ks)
    if LENGTH(ks) = 1 then
        return A[FIRST(ks)]
    else
        return SELECT(A[FIRST(ks)],REST(ks))
    end if
end function
```

Select

dope vector

Require: A :: multidimensional row-major (dope) array
Require: ks :: tuple of indices
  function SELECT(A,ks)
    D $\leftarrow$ A[0]
    index $\leftarrow$ 0
    for $0 \le d < D$ **do**
      index $\leftarrow$ index $\times$ A[1+d] + ks[d]
    end for
    return A[1+D+index]
  end function

Complexity analysis

time

Operations take time proportional to the number of dimensions D, but independent of the size of each dimension. For given D, all operations (size, select, store!) take time in $\Theta(1)$.

space

Iliffe vector
: space overhead proportional to the size of each dimension (worst case, space overhead in $\Theta(N)$)

dope vector
: space overhead proportional to the number of dimensions (for a given dimension, space overhead in $\Theta(1)$)

Multidimensional array (with dope vector) is an example of an implicit data structure