Indexing a linear array as a higher dimensional array

Grady White

July 2022

Formulation:

It is often useful in programming to represent a high dimensional array as an abstraction of a single continuous array. This has the benefit of fast indexing and low storage overhead. This is because the simple arrays are literally represented as such in computer memory. There is no 2-dimensional RAM in a computer and no level of the operating system abstracts this for us. Therefore it is beneficial to treat all arrays in our program as a contiguous block of memory, just like the OS does. With this comes the fast calculation of indices. Since only one index is needed for the linear array, only simple arithmetic operations are needed for calculating the index from a higher dimensional tuple. Formally the indexing can be written as follows.

Choose an N-dimensional array A, with dimension sizes (or maximum indices):

$$S = \{s_0, s_1, s_2, ..., s_{N-1}\} \text{ with } s_i \in S \tag{1}$$

This array can be represented by a linear array a with length:

$$L = \prod_{i=0}^{N-1} S_i \tag{2}$$

There exists an N-tuple which indexes array A such that:

$$(x_0, x_1, x_2, ..., x_{N-1}) \text{ has } x_i \in [0, s_i - 1] \tag{3}$$

Therefore the index to the array a can be written as:

$$I = \sum_{i=0}^{N-1} \left(\prod_{j=i+1}^{N-1} s_j \right) x_i \tag{4}$$

The N-tuple can also be extracted from the set S and the linear index I:

$$x_i = \left(\left(I - \sum_{j=i+1}^{N-1} \left(\prod_{k=j+1}^{N-1} s_k \right) x_j \right) \prod_{j=i+1}^{N-1} s_j^{-1} \right) \bmod s_i \tag{5}$$

There exist identity elements for summation and multiplication such that:

$$\sum_{i >= N-1}^{N-1} = 0 \text{ and } \prod_{i >= N-1}^{N-1} = 1 \tag{6}$$

Examples:

It can be useful to define specific common cases to further increase the speed of these algorithms. It may be beneficial to leave them abstract if many different high dimensional arrays are necessary, but the two most common arrays are 2-D and 3-D.

2-Dimensions:

Consider a 2-dimensional array A with $S = \{3, 3\}$ and a 1-dimensional array a with $L = 9$ The below table displays the result of Eq.4 for all 2-tuples (x, y)

TABLE I

x \ y	0	1	2
0	0	1	2
1	3	4	5
2	6	7	8

A succinct solution to the provided index equation for the 2-tuple (x, y) and $S = \{s_x, s_y\}$:

$$I = x * s_y + y \tag{7}$$

The reverse index equation to obtain the 2-tuple (x, y) using I and $S = \{s_x, s_y\}$:

$$(x, y) = (\frac{I - y}{s_y} \bmod s_z, I \bmod s_y) \tag{8}$$

3-Dimensions:

A succinct solution to the provided index equation for the 3-tuple (x, y, z) and $S = \{s_x, s_y, s_z\}$:

$$I = x * s_y * s_z + y * s_z + z \tag{9}$$

The reverse index equation to obtain the 3-tuple (x, y, z) using I and $S = \{s_x, s_y, s_z\}$:

$$(x, y, z) = (\frac{I - y * s_z - z}{s_y * s_z} \bmod s_x, \frac{I - z}{s_z} \bmod s_y, I \bmod s_z) \tag{10}$$