Article **Talk**

Read **Edit** **View history**

Search

Sparse array

From Wikipedia, the free encyclopedia

> ⚠ **This article has multiple issues.** Please help **improve it** or   [hide]
> discuss these issues on the **talk page**.
>
> - This article **may be too technical** for most readers to
>   **understand**. *(February 2012)*
> - This article **does not cite** any **references or sources**. *(October
>   2011)*

In computer science, a **sparse array** is an array in which most of the elements have the default value (usually 0 or null). The occurrence of zero-value elements in a large array is inefficient for both computation and storage. An array in which there is a large number of zero elements is referred to as being sparse.

In the case of sparse arrays, one can ask for a value from an "empty" array position. If one does this, then for an array of numbers, a value of zero should be returned, and for an array of objects, a value of null should be returned.

A naive implementation of an array may allocate space for the entire array, but in the case where there are few non-default values, this implementation is inefficient. Typically the algorithm used instead of an ordinary array is determined by other known features (or statistical features) of the array. For instance, if the sparsity is known in advance or if the elements are arranged according to some function (e.g., the elements occur in blocks). A heap memory allocator in a program might choose to store regions of blank space in a linked list rather than storing all of the allocated regions in, say a bit array.

Contents [hide]

Representation [edit]

Sparse Array can be represented as

```
Sparse_Array[{pos1 -> val1, pos2 -> val2,...}]
```
 or
```
Sparse_Array[{pos1, pos2,...} -> {val1, val2,...}]
```

which yields a sparse array in which values val_i appear at positions pos_i.

Sparse Array as Linked List [edit]

An obvious question that might be asked is why we need a linked list to represent a sparse array if we can represent it easily using a normal array. The answer to this question lies in the fact that while representing a sparse array as a normal array, a lot of space is allocated for zero or null elements. For example, consider the following array declaration:

```
double arr[1000][1000]
```

When we define this array, enough space for 1,000,000 doubles is allocated. If each double requires 8 bytes of memory, this array will require 8 million bytes of memory. Because this is a sparse array, most of its elements will have a value of zero (or null). Hence, defining this array will soak up all this space and waste memory (compared to an array in which memory has been allocated only for the nonzero elements). An effective way to overcome this problem is to represent the array using a linked list which requires less memory as only elements having non-zero value are stored. This involves a time-space trade-off: though less memory is used, average access and insertion time becomes linear in the number of elements stored because the previous elements in the list must be traversed to find the desired element. A normal array has constant access and insertion time.

A sparse array as a linked list contains nodes linked to each other. In a one-dimensional sparse array, each

node includes the non-zero element's "index" (position), the element's "value", and a node pointer "next" (for linking to the next node). Nodes are linked in order as per the index. In the case of a two-dimensional sparse array, each node contains a row index, a column index (which together give its position), a value at that position and a pointer to the next node.

See also [edit]

- Sparse matrix

External links [edit]

- Boost sparse vector class &
- Rodolphe Buda, "Two Dimensional Aggregation Procedure: An Alternative to the Matrix Algebraic Algorithm", *Computational Economics*, 31(4), May, pp.397–408, 2008. &

v · t · e	Data structures	[show]

Categories: Arrays