Subject: Re: Modifying an array while conserving memory
Posted by Craig Markwardt on Fri, 24 May 2002 03:03:36 GMT
View Forum Message <> Reply to Message

Randall Skelton <rhskelto@atm.ox.ac.uk> writes:

> Hi all,
>
> I have a large array and I would like to 'insert' some data into the
> middle of it. Imagine an array of 1000 points and having 100 points to
> insert beginning at index 500 (the resulting array will have 1100 points).
> Typically, I do not know the length of data I wish to insert until after
> 'a' is defined.
>
> a = findgen(1000)
> b = randomu(seed,100)
> c = fltarr(1100) ; seems wasteful to use more memory
> c[0:499] = a[0:499]
> c[500:599] = b
> c[600:1099] = a[500:999]
>
> In reality, 'a' is of order 2e7 so I would like to avoid making
> multiple copies of it. Does anyone have any suggestions regarding the
> most memory efficient way of doing this?

Hi Randall--

At one time I devised a set of functions called ARRINSERT and
ARRDELETE, in which I tried to do insertion and deletion operations as
efficiently as I thought possible.

I don't think I was able to get around your concern of necessarily
allocating at least as much storage as the original data. What is
really needed at the IDL internal level, is the ability to **augment**
the storage of an existing array efficiently. I tend to do that alot,
say when I'm building a list element by element.

Also, to this day I have strong doubts whether the TEMPORARY()
operator actually saves memory, *during* the operation. [ Of course
it saves memory afterwards since the variable's memory is released. ]

Craig

http://cow.physics.wisc.edu/~craigm/idl/idl.html (under arrays)

--
---------- ----------

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---