Subject: Re: Filling an array
Posted by davidf on Thu, 18 Jan 2001 01:18:58 GMT
View Forum Message <> Reply to Message

Pavel A. Romashkin (pavel.romashkin@noaa.gov) writes:

> If I have
>
> a = findgen(10)
> b = fix(100* randomu(10, 10))
> ; N_elements(a) is equal to n_elements(b)
> c = findgen(total(b))
>
> how can I fill C with values from A using B as a running index, so that
>
> c[0 : b[0]-1] = a[0]
> c[b[0] : b[0]+b[1]-1] = a[1]
>
> etc, without looping through "n_elements(b)-1" iterations?
> I have a fast solution with a loop and indexing using total(/cumulative)
> and a very slow one with loop and replicate, but I can't come up with a
> loop-free one.

Totally impossible. :-(

Cheers,

P.S. Let's just say that usually gets the juices going
on the usual suspects, and I figured you could use the
help. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Filling an array
Posted by Craig Markwardt on Thu, 18 Jan 2001 05:37:24 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) writes:

> Pavel A. Romashkin (pavel.romashkin@noaa.gov) writes:
>
>> If I have

>>
>> a = findgen(10)
>> b = fix(100* randomu(10, 10))
>> ; N_elements(a) is equal to n_elements(b)
>> c = findgen(total(b))
>>
>> how can I fill C with values from A using B as a running index, so that
>>
>> c[0 : b[0]-1] = a[0]
>> c[b[0] : b[0]+b[1]-1] = a[1]
>>
>> etc, without looping through "n_elements(b)-1" iterations?
>> I have a fast solution with a loop and indexing using total(/cumulative)
>> and a very slow one with loop and replicate, but I can't come up with a
>> loop-free one.
>
> Totally impossible. :-(
>
> Cheers,
>
> P.S. Let's just say that usually gets the juices going
> on the usual suspects, and I figured you could use the
> help. :-)

Juices or not, I don't see how this can be done without a loop. Since the segments specified by B can be of different sizes, I think you are stuck. However, as I've said in the past: loops aren't bad! If you can get enough work done in a single iteration then loops are fine.

Craig

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

Subject: Re: Filling an array
Posted by Pavel A. Romashkin on Thu, 18 Jan 2001 17:03:32 GMT
View Forum Message <> Reply to Message

Thanks David and Craig. If Craig says "no", this means something. Although I'd wait with the verdict until I hear from JD :-) After all, the loop I have with Total, although it goes through all elements of B, is taking only 0.03 s on my machine for B with ~2500 points converted to C with ~50k points, which is acceptable since it is not executed many times repeatedly. In contrast with looping using Replicate that was

taking 5.5 s :-(

Cheers,
Pavel

"Pavel A. Romashkin" wrote:
>
> If I have
>
> a = findgen(10)
> b = fix(100* randomu(10, 10))
> ; N_elements(a) is equal to n_elements(b)
> c = findgen(total(b))
>
> how can I fill C with values from A using B as a running index, so that
>
> c[0 : b[0]-1] = a[0]
> c[b[0] : b[0]+b[1]-1] = a[1]
>
> etc, without looping through "n_elements(b)-1" iterations?
> I have a fast solution with a loop and indexing using total(/cumulative)
> and a very slow one with loop and replicate, but I can't come up with a
> loop-free one.
>
> Thank you,
> Pavel

Subject: Re: Filling an array
Posted by John-David T. Smith on Thu, 18 Jan 2001 20:28:54 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" wrote:
>
> Thanks David and Craig. If Craig says "no", this means something.
> Although I'd wait with the verdict until I hear from JD :-) After all,
> the loop I have with Total, although it goes through all elements of B,
> is taking only 0.03 s on my machine for B with ~2500 points converted to
> C with ~50k points, which is acceptable since it is not executed many
> times repeatedly. In contrast with looping using Replicate that was
> taking 5.5 s :-(
>
> Cheers,
> Pavel
>
> "Pavel A. Romashkin" wrote:
>>
>> If I have

>>
>> a = findgen(10)
>> b = fix(100* randomu(10, 10))
>> ; N_elements(a) is equal to n_elements(b)
>> c = findgen(total(b))
>>
>> how can I fill C with values from A using B as a running index, so that
>>
>> c[0 : b[0]-1] = a[0]
>> c[b[0] : b[0]+b[1]-1] = a[1]
>>
>> etc, without looping through "n_elements(b)-1" iterations?
>> I have a fast solution with a loop and indexing using total(/cumulative)
>> and a very slow one with loop and replicate, but I can't come up with a
>> loop-free one.

Despite the fact that getting a job and writing a thesis should be my
foremost priorities....

 tmp=histogram(total(b,/CUMULATIVE)-1,/BINSIZE,MIN=0,REVERSE_ INDICES=ri)
c=a[ri[0:n_elements(ri)-n_elements(b)-2]-ri[0]]

JD

P.S. Let's just say (TM) we all knew it had to use histogram.

P.P.S. A loop free solution is not guaranteed to be fastest in all
cases.   It *is* guaranteed to elicit various
histogram/median/rebin/reform/## curses from newsgroup readers.

Subject: Re: Filling an array
Posted by davidf on Thu, 18 Jan 2001 20:52:16 GMT
View Forum Message <> Reply to Message

JD Smith (jdsmith@astro.cornell.edu) writes:

> P.S. Let's just say (TM) we all knew it had to use histogram.

Oh, well, yes, with a histogram. I though Pavel meant
a way *without* using a histogram. :-(

Cheers,

David

--
David Fanning, Ph.D.

Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Filling an array
Posted by Craig Markwardt on Thu, 18 Jan 2001 21:40:47 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@astro.cornell.edu> writes:
> tmp=histogram(total(b,/CUMULATIVE)-1,/BINSIZE,MIN=0,REVERSE_ INDICES=ri)
> c=a[ri[0:n_elements(ri)-n_elements(b)-2]-ri[0]]
>

Bravo. I guess the juices just weren't flowing last night :-)

Craig

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

Subject: Re: Filling an array
Posted by Pavel A. Romashkin on Fri, 19 Jan 2001 00:43:55 GMT
View Forum Message <> Reply to Message

I knew it. What the heck will I write to the newsgroup the next time for? I will just write directly to JD :-) I will give it a shot timing wise tomorrow, but I am sure it will be faster.
This one was the last drop in the bucket. I am reading and practicing with the darn Histogram thing tomorrow until I can write my entire code with a long single call to Histogram.

Cheers,
Pavel

P.S. Lets just say (TM) that I tried to rig up Histogram for this, but was distracted before I got too far. Not that I am saying I'd come up with a solution, if I wasn't :-(

JD Smith wrote:
>
> "Pavel A. Romashkin" wrote:

>>
>> Thanks David and Craig. If Craig says "no", this means something.
>> Although I'd wait with the verdict until I hear from JD :-) After all,
>> the loop I have with Total, although it goes through all elements of B,
>> is taking only 0.03 s on my machine for B with ~2500 points converted to
>> C with ~50k points, which is acceptable since it is not executed many
>> times repeatedly. In contrast with looping using Replicate that was
>> taking 5.5 s :-(
>>
>> Cheers,
>> Pavel
>>
>> "Pavel A. Romashkin" wrote:
>>>
>>> If I have
>>>
>>> a = findgen(10)
>>> b = fix(100* randomu(10, 10))
>>> ; N_elements(a) is equal to n_elements(b)
>>> c = findgen(total(b))
>>>
>>> how can I fill C with values from A using B as a running index, so that
>>>
>>> c[0 : b[0]-1] = a[0]
>>> c[b[0] : b[0]+b[1]-1] = a[1]
>>>
>>> etc, without looping through "n_elements(b)-1" iterations?
>>> I have a fast solution with a loop and indexing using total(/cumulative)
>>> and a very slow one with loop and replicate, but I can't come up with a
>>> loop-free one.
>
> Despite the fact that getting a job and writing a thesis should be my
> foremost priorities....
>
> tmp=histogram(total(b,/CUMULATIVE)-1,/BINSIZE,MIN=0,REVERSE_ INDICES=ri)
> c=a[ri[0:n_elements(ri)-n_elements(b)-2]-ri[0]]
>
> JD
>
> P.S. Let's just say (TM) we all knew it had to use histogram.
>
> P.P.S. A loop free solution is not guaranteed to be fastest in all
> cases. It *is* guaranteed to elicit various
> histogram/median/rebin/reform/## curses from newsgroup readers.