Subject: Re: Large Numbers
Posted by pgrigis on Fri, 06 Feb 2009 23:35:11 GMT
View Forum Message <> Reply to Message

Well, if you deal with very large numbers,
you can do all the computations
with the logarithm of the numbers.

Simple, no?

Ciao,
Paolo

David Fanning wrote:
> Folks,
>
> I made a big mistake and signed up for an Applied Statistics
> class this semester. Now I pretty much spend every free
> waking moment doing stats homework. :-(
>
> Anyway, for lunch today I decided to grab a sandwich and
> give my youngest some support by calculating how many
> girls he had to ask out to have an 80% chance of getting
> a date for Saturday night.
>
> I made some conservative assumptions (I learned later
> my ideas about the college social scene apply more to the
> 1970s than they do to today), and off I went writing a
> couple of short IDL programs to do the calculations for
> the Binomial and Geometry Distributions, etc. All pretty
> straightforward.
>
> But then I started getting screwy results. (This, in itself,
> is not all that unusual in this particular class. In fact, I've
> begun to consider it something of a minor miracle if I'm within
> an order of magnitude of the right answer.) But even I know
> that negative probabilities don't show up until the second
> semester. What in the world!?
>
> It turns out that the recursive function I naively wrote to
> process a factorial calculation was overflowing my long
> integers, even with a simple calculation like 20! (twenty
> factorial). Yowser!
>
> Now, of course, the formula I was using has a large
> factorial number divided by another large factorial
> number, so the *actual* number I wanted to use in the

> calculation is not that big. But it begs the question:
> what strategy do computer scientists use to deal with
> one very, very big number divided by another very, very
> big number?
>
> I've solved my immediate problem for my little toy problem
> by using LONG64 variables. But this can't be the right solution.
> Does anyone know?
>
> Cheers,
>
> David
>
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by Chris[6] on Sat, 07 Feb 2009 00:49:16 GMT
View Forum Message <> Reply to Message

On Feb 6, 1:12 pm, David Fanning <n...@dfanning.com> wrote:
> Folks,
>
> I made a big mistake and signed up for an Applied Statistics
> class this semester. Now I pretty much spend every free
> waking moment doing stats homework. :-(
>
> Anyway, for lunch today I decided to grab a sandwich and
> give my youngest some support by calculating how many
> girls he had to ask out to have an 80% chance of getting
> a date for Saturday night.
>
> I made some conservative assumptions (I learned later
> my ideas about the college social scene apply more to the
> 1970s than they do to today), and off I went writing a
> couple of short IDL programs to do the calculations for
> the Binomial and Geometry Distributions, etc. All pretty
> straightforward.
>
> But then I started getting screwy results. (This, in itself,
> is not all that unusual in this particular class. In fact, I've
> begun to consider it something of a minor miracle if I'm within
> an order of magnitude of the right answer.) But even I know

> that negative probabilities don't show up until the second
> semester. What in the world!?
>
> It turns out that the recursive function I naively wrote to
> process a factorial calculation was overflowing my long
> integers, even with a simple calculation like 20! (twenty
> factorial). Yowser!
>
> Now, of course, the formula I was using has a large
> factorial number divided by another large factorial
> number, so the *actual* number I wanted to use in the
> calculation is not that big. But it begs the question:
> what strategy do computer scientists use to deal with
> one very, very big number divided by another very, very
> big number?
>
> I've solved my immediate problem for my little toy problem
> by using LONG64 variables. But this can't be the right solution.
> Does anyone know?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Java, for example, has a bigInteger class, which internally represents
a big integer as an array of 32 bit integers - something like
decimal_equivalent = sum(array[i] * (2^32)^i)

http://developer.classpath.org/doc/java/math/BigInteger-sour ce.html

chris

Subject: Re: Large Numbers
Posted by David Fanning on Sat, 07 Feb 2009 00:50:21 GMT
View Forum Message <> Reply to Message

Paolo writes:

> Well, if you deal with very large numbers,
> you can do all the computations
> with the logarithm of the numbers.

>
> Simple, no?

Uh, probably, if I can find those dusty books
down in the basement. :-(

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by David Fanning on Sat, 07 Feb 2009 00:54:38 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Uh, probably, if I can find those dusty books
> down in the basement. :-(

I found my slide rule. Maybe that will help.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by pgrigis on Sat, 07 Feb 2009 01:14:38 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> David Fanning writes:
>
>> Uh, probably, if I can find those dusty books
>> down in the basement. :-(
>

> I found my slide rule. Maybe that will help.

For instance, you can easily find that
1000!=4.023*10E2567

How?

the trick is that
gamma(x+1)=factorial(x)

therefore, alog(factorial(1000))=lngamma(1001)=2567.6046
10^0.6046=4.023

1000!=4.023*10E2567

Ciao,
Paolo

>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by Kenneth P. Bowman on Sat, 07 Feb 2009 02:57:07 GMT
View Forum Message <> Reply to Message

In article <MPG.23f671b522eba5ef98a5f6@news.giganews.com>,
David Fanning <news@dfanning.com> wrote:

> Now, of course, the formula I was using has a large
> factorial number divided by another large factorial
> number, so the *actual* number I wanted to use in the
> calculation is not that big. But it begs the question:
> what strategy do computer scientists use to deal with
> one very, very big number divided by another very, very
> big number?

If you are dividing them, you don't really want to use
integers, do you? That's what floats (and doubles) are for.

Ken Bowman

Subject: Re: Large Numbers
Posted by David Fanning on Sat, 07 Feb 2009 03:53:10 GMT
View Forum Message <> Reply to Message

Paolo writes:

> For instance, you can easily find that
> 1000!=4.023*10E2567

Humm. Maybe you can easily find it. All I found was
an old notebook with spider egg casings on the cover,
and what looks like the remains of an anchovy pizza
spilled on the phone number of that blonde in the
bikini who ended up with my roommate's older brother. :-(

Cheers,

David

P.S. Thanks for the tips. Who knew logarithms could
be so useful! :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by David Fanning on Sat, 07 Feb 2009 03:59:46 GMT
View Forum Message <> Reply to Message

Kenneth P. Bowman writes:

> If you are dividing them, you don't really want to use
> integers, do you? That's what floats (and doubles) are for.

Humm. You might have a point there.

Ok, how about a contest. Given two integers, x and n,
with x le n, what is the most efficient way to evaluate
this expression:

n!/(x! * (n-x)!)

The usual prize (as my children know well, a hug
and a kiss) to the winner. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by pgrigis on Sat, 07 Feb 2009 04:06:11 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Kenneth P. Bowman writes:
>
>> If you are dividing them, you don't really want to use
>> integers, do you? That's what floats (and doubles) are for.
>
> Humm. You might have a point there.
>
> Ok, how about a contest. Given two integers, x and n,
> with x le n, what is the most efficient way to evaluate
> this expression:
>
> n!/(x! * (n-x)!)

You know about the function "factorial", right?

Ciao,
Paolo

>
> The usual prize (as my children know well, a hug
> and a kiss) to the winner. :-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by David Fanning on Sat, 07 Feb 2009 04:13:06 GMT
View Forum Message <> Reply to Message

Paolo writes:

> You know about the function "factorial", right?

Holy smokes! No. And BINOMIAL is right there beside it!
Sheesh, I've spent all day re-inventing the wheel.

Well, at least no one can accuse me of not knowing statistics
from first principles. ;-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Large Numbers
Posted by Jeremy Bailin on Tue, 10 Feb 2009 04:28:27 GMT
View Forum Message <> Reply to Message

On Feb 6, 6:35 pm, Paolo <pgri...@gmail.com> wrote:
> Well, if you deal with very large numbers,
> you can do all the computations
> with the logarithm of the numbers.
>
> Simple, no?
>
> Ciao,
> Paolo
>
> David Fanning wrote:
>> Folks,
>
>> I made a big mistake and signed up for an Applied Statistics
>> class this semester. Now I pretty much spend every free
>> waking moment doing stats homework. :-(
>
>> Anyway, for lunch today I decided to grab a sandwich and
>> give my youngest some support by calculating how many
>> girls he had to ask out to have an 80% chance of getting

>> a date for Saturday night.
>
>> I made some conservative assumptions (I learned later
>> my ideas about the college social scene apply more to the
>> 1970s than they do to today), and off I went writing a
>> couple of short IDL programs to do the calculations for
>> the Binomial and Geometry Distributions, etc. All pretty
>> straightforward.
>
>> But then I started getting screwy results. (This, in itself,
>> is not all that unusual in this particular class. In fact, I've
>> begun to consider it something of a minor miracle if I'm within
>> an order of magnitude of the right answer.) But even I know
>> that negative probabilities don't show up until the second
>> semester. What in the world!?
>
>> It turns out that the recursive function I naively wrote to
>> process a factorial calculation was overflowing my long
>> integers, even with a simple calculation like 20! (twenty
>> factorial). Yowser!
>
>> Now, of course, the formula I was using has a large
>> factorial number divided by another large factorial
>> number, so the *actual* number I wanted to use in the
>> calculation is not that big. But it begs the question:
>> what strategy do computer scientists use to deal with
>> one very, very big number divided by another very, very
>> big number?
>
>> I've solved my immediate problem for my little toy problem
>> by using LONG64 variables. But this can't be the right solution.
>> Does anyone know?
>
>> Cheers,
>
>> David
>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
>

What I've run into a few times lately is taking sums and differences
of very large numbers. Logarithms are not so useful in those cases...
anyone have any general useful techniques (beyond the usual "re-phrase

your equations to only calculate the differences between sums instead
of the sums themselves")?

-Jeremy.

Subject: Re: Large Numbers
Posted by Michael Galloy on Tue, 10 Feb 2009 17:28:40 GMT
View Forum Message <> Reply to Message

Jeremy Bailin wrote:
> What I've run into a few times lately is taking sums and differences
> of very large numbers. Logarithms are not so useful in those cases...
> anyone have any general useful techniques (beyond the usual "re-phrase
> your equations to only calculate the differences between sums instead
> of the sums themselves")?

Absent techniques for the particular problem, you could always use Ron
Kneusel's BigNum (for integers) and Arbitrary precision floating point
packages. They are on the ITT VIS code contrib library.

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist
