

Re: Fastcode IntToStr 0.15 B&V

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.language.basm/2005-09/msg00213.html>

- *From:* "Avatar Zondertau" <avatarzt@xxxxxxxxxx> (please reply to newsgroup)>
 - *Date:* 5 Sep 2005 22:52:49 -0700
-

>> This causing the
>> smaller intervals to be weighed more strongly even though the
>> subbenchmark is not. I think this behaviour is desirable since i
>> think smaller values are more likely.
>
> Why do we know that smaller values are more likely?

There are more large numbers than small numbers, so if one uses as many large numbers as small numbers, the individual large numbers get called less often. I think this is the most likely usage scenario.

If the benchmark would rely test with a uniform distribution the small numbers would relatively get next to no attention, which results in the cases of numbers of shorter length not really mattering for the result. Remember that there are 10 times as many numbers with 9 digits as with 8, 100 times more with 9 than with 7 and so on.

If we want all different lengths give equal weight we should, i think, use a distribution where the likelihood of a number being called is proportional to it's logarithm.

I think the distribution of numbers has quite some effect on which approach is optimal.

> Is the runtime of functions independent from the parameter value?

No! Larger numbers take more time as they require more digits to be calculated and written.

>>> 3) The benchmark accuracy must be better than 1%.
>>
>> How does one calculate this number? Is it the benchmark spread?
>
> Yes. I have two ways of testing it. One button runs the same function
> 20 times and calculates the spread. Another button runs through all
> functions a number of times. It uses the same calculation of spread.
> Then we can manually perform benchmarks and benchmarks all. Again the
> same calculation of spread is automatically used.

So basically i could just add a spinedit next to the benchmark buttons to specify a number of times?

> > > 4) The benchmark must put maximum pressure on the branch predictor
> >
> > I think my benchmark does this by using random numbers instead of
> > sequentially going over the range. The sequence is fixed however, so
> > each function gets the same value (RandSeed is initialized to a
> > constant).
>
> This sounds ok. Most important is that the sequence is long enough to
> overflow the branch target buffer.

Approximately how long do you think that would be?

• *Follow-Ups:*

- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Dennis*

• *References:*

- ◆ *Fastcode IntToStr 0.15 B&V*
◇ *From: Avatar Zondertau*
- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Dennis*
- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Avatar Zondertau*
- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Dennis*
- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Avatar Zondertau*
- ◆ *Re: Fastcode IntToStr 0.15 B&V*
◇ *From: Dennis*

- Prev by Date: *attn: issie – highly pleasant premium nntp access – ar avve uz ofce – (1/1)*
- Next by Date: *Re: IntToStr challenge rules*
- Previous by thread: *Re: Fastcode IntToStr 0.15 B&V*
- Next by thread: *Re: Fastcode IntToStr 0.15 B&V*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*