

Re: Algorithm wanted

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2004-03/0620.html>

From: CBFalconer (cbfalconer_at_yahoo.com)

Date: 03/08/04

Date: Mon, 08 Mar 2004 00:48:30 GMT

*** top-posting fixed ***

Kurt Van Samang wrote:

> *"Kurt Van Samang" schreef in bericht*

> >

> > *I need an algorithm that will do more or less this. Let's say
> > we have an array with some numbers, size is variable. Now I
> > would like to throw away all values that aren't in order. Some
> > examples:*

> >

> > *5-1-2-3-4 will generate 1-2-3-4*

> > *5-1-6-7-8 will give 5-6-7-8*

> > *10-5-6-21-35-42 will give 10-21-35-42 or 5-6-21-35-42, doesn't*

> > *matter that much, as long as the remaining items are in order*

> >

> *I want to thank everyone who answered. I'm sorry that I wasn't
> quite clear about whether I wanted the longest row or not. What
> I actually want is to detect some "intruders" that incorrectly
> where inserted into the row. I'm working on a computer graphics
> project in which I take evenly-spaced samples from a polyline
> and transfer them to a polyline in the same neighbourhood in
> the next frame. But sometimes some of them arrive at a wrong
> position. Since they all get an id at start I wanted to detect
> those that arrived on a incorrect position. So it doesn't
> really have to be the longest row.*

> >

> *Let's change the example 10-5-6-21-35-42 to 18-5-6-21-25-29,*

> *then I would prefer 18-21-25-29 cause it seems like 5 and 6*

> *don't belong there. Anyway, it doesn't have to be 100% correct,*

> *so I could live with a result of 5-6-21-25-29, as long as they*

> *are in order. Of course in the example 5-1-2-3-4 I'm not quite*

> *happy with just 5, cause it's quite obvious that 5 doesn't*

> *belong there. So I guess I'm gonna try the longest ascending*

> *subsequence algorithm.*

Please do not top-post. I fixed this one.

comp.programming: Re: Algorithm wanted

This still isn't a clear specification, except for longest ascending ... :-)

This sounds vaguely like the methods I used years ago to calibrate blood testing machinery. We fed it known samples over a range, and recorded the results. Now we did a least squares fit to a 3rd order polynomial, and rescanned the recorded input data for anomalous errors. The worst point (if any) was discarded, and the process repeated, until either there were no more anomalous inputs or the input data was too sparse. If the data was too sparse the system screamed for the operator.

The results were highly accurate and repeatable.

--

Chuck F (cbfalconer@yahoo.com) (cbfalconer@worldnet.att.net)
Available for consulting/temporary embedded and systems.
<<http://cbfalconer.home.att.net>> USE worldnet address!