

Re: Using hashes to sort number sequences

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2004-05/1319.html>

From: Bob Walton (*invalid-email_at_rochester.rr.com*)

Date: 05/13/04

Date: Thu, 13 May 2004 03:07:13 GMT

Martin Foster wrote:

...

> *I have two files: a.txt & b.txt*

>

> *a.txt=*

> *191_6_270328 T1 4 10 19 34 55 72 88 116 157 200 280 332 388 451 756 4*

> *0 5 0 4 0 6 2 6 2 8 0*

> *191_6_270328 T2 4 9 17 22 34 56 83 112 146 181 266 320 376 431 665 3 0*

...

> *b.txt=*

> *191_6_9908682 T1 4 8 14 25 41 60 83 115 153 190 276 321 374 437 694 4*

> *0 4 0 4 0 6 0 4 0 8 0*

> *191_6_9908682 T2 4 10 19 30 44 64 92 122 155 198 285 338 394 446 739 4*

> *0 5 0 4 0 6 0 8 0 8 2*

...

> *Each file contains in the first column an identifier, I call it \$name.*

> *The 2nd column contains an entry T1 or T2 or T3 ... until T6.*

> *After these two columns each row contains a number sequence.*

>

> *What I would like to do is to read file a.txt, six lines at a time*

> *(from T1 to T6)*

> *and search for similar number sequences in file b.txt.*

> *The number sequences in file b.txt must also be within each block of*

> *six lines,*

> *but they can be in any order.*

Why don't you just sort (using the Unix or maybe even the Win32 sort command) the two files, and then, using Perl, read and compare from the two sorted files? Or maybe the `-u` switch on Unix's sort could give you what you want in one go. Or maybe (if the data for matching lines is all the same), after the sorts, use diff to do the compare, and just process the output of diff with Perl? Or if there is something in the data which indicates if it from INFILE1 versus INFILE2, the files could be concatenated, sorted, and processed as one file (I don't think that last method would have any advantages).

comp.lang.perl.misc: Re: Using hashes to sort number sequences

That sort (punny, huh?) of method will avoid reading your \$infile2 many hundreds of thousands of times, which will take almost forever.

BTW, you would need to either close and reopen the file in your inner loop, or seek() it back to the beginning every time you go through the outer loop. Also recognize that your while(<INFILE1>) and while(<INFILE2>) constructions will read a record from the corresponding file and place it into \$_. You are discarding that data, so you are really reading data 7 records at a time, discarding the first of each chunk of 7.

HTH.

...

> *Martin.*

--

Bob Walton

Email: <http://bwalton.com/cgi-bin/emailbob.pl>