<<Perl   >>

<<Perl   >>

13   ISBN   9787564138882

10   ISBN   7564138882

         2013-1

                              PDF

            http://www.tushu007.com

<<Perl   >>

Page 2

Perl   (   )(2)

Perl   CPAN   ……   Perl
1991   Perl

<<Perl   >>

Randal L.Scbwartz   Brian d Foy   Tom
Pboenix   Randal L. Schwartz
                              Perl   Perl   Perl   Perl
           O'Reilly
brian d foy   Perl   The PerlReview   Perl

       Perl   Perl   Perl   EffectivePerl Programming   Addison
—Wesley
Tom Phoenix   StonehengeConsulting Services   Perl
           Usenet   comp.lang.perl.misc   complang.pefl.moderated
     Perl   Perl

<<Perl   >>

Sorting Efficiently As the Professor tries to maintain the community computing facility (built entirely out of bamboo, coconuts, and pineapples, and powered by a certified Perl-hacking monkey), he continues to discover that people are leaving entirely too much data on the single monkey-powered filesystem, so he decides to print a list of offenders.The Professor has written a subroutine called ask_monkey_about, which, given a cast-away's name, returns the number of pineapples of storage they use. We have to ask the monkey because he's in charge of the pineapples. An initial naive approach to find the offenders from greatest to least might be something like:In theory, this would be fine. For the first pair of names (Gilligan and Skipper), we ask the monkey "How many pineapples does Gilligan have?" and "How many pineapples does Skipper have?" We get back two values from the monkey and use them to order Gilligan and Skipper in the final list. However, at some point, we have to compare the number of pineapples that Gilligan has with another castaway as well. For example, suppose the pair is Ginger and Gilligan. We ask the monkey about Ginger, get a number back, and then ask the monkey about Gilligan... again. This will probably annoy the monkey a bit, since we already asked. But we need to ask for each value two, three, or maybe even four times just to put these ven values into order. This can be a problem because it irritates the monkey. How do we keep the number of monkey requests to a minimum? Well, we can build a table first. We use a map with seven inputs and seven outputs, turning each castaway item into a separate array reference, with each referenced array consisting of the cast-away name and the pineapple count reported by the monkey.

<<Perl   >>

Perl   (   )(2)   Perl
    Perl   Perl

<<Perl   >>

PDF

:http://www.tushu007.com