

Scripting Languages G22.3033-002 Summer 2008 hw04

Assigned Th 6/12/2008, due Fr 6/20 at 9pm. 50 points.

<http://www.cs.nyu.edu/courses/summer08/G22.3033-002/>

Homework instructions

Homeworks are due on Fridays at 9pm. This deadline will be strictly enforced.

Email your answers to Robert Soulé robert.soule@gmail.com. Please put your solutions to VBA programming problems in a powerpoint presentation. For all other questions (including programming problems in other languages), just send a simple text file, such as what you get when using Emacs, Vi, Notepad, or the "save as text only" feature in Word.

Please make sure that your code works with the compilers and tools installed at CIWW. Specifically, please test:

JAVASCRIPT

Your JavaScript code must run with **both** Mozilla's Firefox browser **and** with Microsoft's Internet Explorer browser.

PHP Your PHP code must run with PHP 5.1.6 on the CIMS web servers, see <http://www.cims.nyu.edu/systems/userservices/webhosting/>.

PERL Your Perl code must run with Perl 5.8.8 for Linux/x86, for example on doowop1 (see <http://www.cims.nyu.edu/systems/resources/computeservers/>).

VBA Your VBA code must run with Microsoft Office 2003 for Windows on the machines in the labs CIWW 502 or CIWW 624.

Reading assignments

Read for lecture on 6/19:

W3Schools PHP Tutorial. Read the part "PHP Basics", which should start with "PHP Intro" and end with "PHP \$_POST". Available online at <http://www.w3schools.com/php/>. If you don't yet know HTML, read the tutorial for HTML first, at <http://www.w3schools.com/html/>.

Concept questions

hw04-1 Weak, strong, static, and dynamic typing

(5+5 = 10 points)

- Dynamic typing is not the same as weak typing. Explain the difference.
- Look at the picture categorizing language type systems along two axes in the 6/12/2008 lecture slides. Is VBA placed correctly in that picture? Explain.

hw04-2 Context in Perl

(4+4+4 = 12 points)

- Consider the following Perl function.

```

sub isprime {
    my ($num, $f, @factors) = ($_[0], 2);
    while (1 != $num) {
        if (0 == $num % $f) {
            return $f == $num unless wantarray;
            push @factors, $f;
            $num /= $f;
        } else {
            $f++;
        }
    }
    return @factors;
}

```

Give four examples for calling this function: with a prime or a non-prime, in a list context or a scalar context.

- b. Consider the following Perl function.

```

sub map_block (&@) {
    my ($block, @listin) = @_;
    my @listout = ();
    push @listout, &$block() for (@listin);
    return @listout;
}

```

Explain where and how this function uses the default variable `$_`.

- c. Show how to use function `map_block` above to add 5 to every element of a list. For example, your code should turn the list 1,2,3 into the list 6,7,8.

Programming exercises

hw04-3 Bidirectional mapping (Perl)

(10 + 4 = 14 points)

- a. Use the object-oriented features of Perl to implement a class `TwoWayMap`. An instance of this class should behave like an associative array that maps keys to values, but should also work in the reverse direction, mapping values back to keys. For example, the following code:

```

use TwoWayMap;
our $e2g = new TwoWayMap;
$e2g->put('apple', 'Apfel');
$e2g->put('pear', 'Birne');
print $e2g->get('apple'), "\n";
print $e2g->rev('Apfel'), "\n";
print $e2g->rev('Birne'), "\n";

```

should print `Apfel, apple, pear`. You can assume that all keys and values are strings.

- b. Improve the constructor of your class `TwoWayMap` so that it also accepts an initialization list. For example, the following code:

```

use TwoWayMap;
our $e2g = new TwoWayMap 'grape' => 'Traube', 'onion' => 'Zwiebel';
print $e2g->get('grape'), "\n";
print $e2g->rev('Traube'), "\n";
print $e2g->rev('Zwiebel'), "\n";

```

should print Traube, grape, onion.

hw04-4 Hierarchical clustering (Perl)

(14 points) Write a Perl script that reads a table and writes a dendrogram specification. Consider this example input:

```

A, 4, 0
B, 8, 5
C, 4, 4

```

There are three rows. Hierarchical clustering starts by putting each row in a cluster by itself. Then, it finds the two clusters with the smallest Euclidian distance. The distance between clusters A and C is $\sqrt{(4-4)**2+(4-0)**2}=4$, and these two clusters are closest to each other. Hierarchical clustering combines them to obtain a partial dendrogram specification (A:4:C). The combined cluster (A:4:C) has two rows and the arithmetic mean $[(4+4)/2, (0+4)/2] = [4, 2]$. Thus, the current set of clusters is:

```

(A:4:C) => 2 x [4, 2]
      B => 1 x [8, 5]

```

Now, hierarchical clustering repeats the same step. There are only two clusters left, so those are automatically the two clusters with the smallest Euclidian distance. The distance between the two remaining clusters is $\sqrt{(8-4)**2+(5-2)**2}=5$. Hierarchical clustering combines them as before, using a weighted average. The set of clusters becomes:

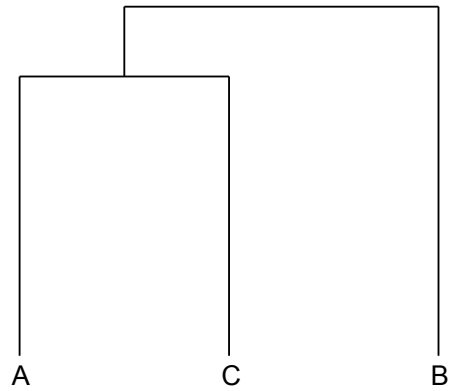
```

((A:4:C):5:B) => 3 x [5.33, 3]

```

Since there is only one cluster left, hierarchical clustering terminates. Your Perl script should print the final dendrogram specification ((A:4:C):5:B) to standard output.

You can then copy-and-paste the output of your Perl script into the VBA dialog box you designed for hw02-4 to visualize the dendrogram as a tree. With appropriate scaling, this is what it looks like:



Here is the link to the example solution from hw02-4:

<http://www.cs.nyu.edu/courses/summer08/G22.3033-002/solutions-hw02-4.ppt>