Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Sandeep Sadanandan

Sommer Semester 2009
Problem Sheet 7
5. Juli 2010

Python For Fine Programmers

There aren't any real question/problems this week to be submitted. But here is a list of mini-projects which you could solve during the following three weeks.

Today being the 5th of July, the date of submission is 26th of July (which is exactly three weeks from now)

The project carries three times the number of points which an assignment would carry.

The list

1. Spell Checker

   Write a spell-checker (and suggesting better spelling) using Python. The checking for spelling part wouldn't be too hard, but also implement functionalities which would suggest the closest matches of the word which is in the file.

   Please see how the tool `ispell` is used in Linux/Unix.

2. Code Beautifier

   Write a Python program to beautify the code written in one of your favourite languages (of course other than python, for python doesn't need much beautification).

   For more details, one could look into `indent(1)` in Linux/Unix.

3. Encrypt / Decrypt

   Implement some of cryptographic algorithms starting from Caeser's code, upto RSA.

   Also create a program which computes, given a message, computes how much expensive each cryptographic algorithm would be to encrypt it.

   (Bonus: Primality testing algorithms are also worth implementing)

4. Handwriting to Image

   Create font from your handwriting.

   - Write on a piece of paper

   - Scan it and get your handwriting (as small png/jpg files)

   - Convert text into an image of the small images created from the png/jpg files

   So, the main idea is to have a stitching algorithm which takes small images and creates them into the big picture which is the handwritten copy of the typed text.

5. Hangman The name explains it all. You can have either or both of the following.

   - Text based
   - Gui based (Tkinter / Pygame)

   Could be Fuuuuuuuuuun!

6. Tic Tac Toe

   Implement Tic Tac Toe

   (if you dare enough, implement Chess (text/gui))

7. Simply Algorithms

   Choose random algorithms and implement some (10-15) of them.

   From : `http://en.wikipedia.org/wiki/List_of_algorithms`

8. Emacs Doctor in Python?

   If you know him/her, nice. If you don't, know him/her.

   And have a duplicate in Python (instead of LISP)

9. Star Locator

   Given a persons date of birth, compute his age and find a star which is that many light years away from earth. (so that the light coming from the star now is as old as he/she is) [There is a list of stars / distance in Wikipedia]

   And then, get the coordinates of the star (from wikipedia) and find out the time and direction to look from earth if one is to see the star.

   (for more details, contact your tutor)

10. ASCII Art?

    Image to ASCII-art.

11. ACM Problems

    Choose a few (10-15) ACM programming contest problems and code them

12. Word Guesser

    Learns the words you type frequently and suggests them when you start typing the beginning of the words.

    (create a small text field; the recommended list of words appear on a panel on the right side, choosable with ctrl-#)

13. Shell Gui

    Create a Shell.

    - Shell - prompt and scrollable screen.
    - Command parser (or direct OS output)

- Command history
- Background running
- Output directions
- Tab-Completion
- Pipes?