Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre de 2014

IIC1103 - Sección 8
Guía de ejercicios

Introducción

Los siguientes ejercicios han sido tomados de `projecteuler.net`. Para revisar el planteamiento original de un problema, puedes ir a `projecteuler.net/problem=X`, remplazando X por el número del problema correspondiente. Los ejercicios están enumerados de la a hasta la i, y entre corchetes se indica la enumeración original de cada problema en `projecteuler.net`.

Ejercicios

a) [problema 3] Los factores primos de 13195 son 5, 7, 13 y 29. ¿Cuál es el factor primo más grande del número 600851475143?

b) [problema 4] Un número capicúa es un número que se lee de igual forma en ambas direcciones. Por ejemplo, el número 23432 es capicúa. El número capicúa más grande que se puede expresar como el producto de dos números de dos dígitos es 9009, pues $9009 = 99 \cdot 91$. ¿Cuál es el número capicúa más grande que se puede expresar como el producto de dos números de tres dígitos?

c) [problema 6] La suma de los cuadrados de los primeros diez naturales es

$$1^2 + 2^2 + \cdots + 10^2 = 385.$$

El cuadrado de la suma de los primeros diez números naturales es

$$(1 + 2 + \cdots + 10)^2 = 3025.$$

Luego, la diferencia entre la suma de los cuadrados de los primeros diez números naturales y el cuadrado de la suma de los primeros diez números naturales es $3025 - 385 = 2640$. Encuentra la diferencia entre la suma de los cuadrados de los primeros cien números naturales y el cuadrado de la suma de los primeros cien números naturales

d) [Problema 9] Un trío pitagórico es una tupla de números (a, b, c) con $a < b < c$ tal que $a^2 + b^2 = c^2$. Existe exactamente un trío pitagórico (a, b, c) tal que $a + b + c = 1000$. Encuentra el producto $a \cdot b \cdot c$ para dicho trío pitagórico.

e) [Problema 14] Sea $f : \mathbb{N} \to \mathbb{N}$ la función definida como

$$f(n) = \begin{cases} n/2 \text{ si } n \text{ es par} \\ 3n + 1 \text{ si } n \text{ es impar.} \end{cases}$$

Dado el número 13, si aplicamos repetidamente la función f hasta llegar a 1, obtenemos la siguiente secuencia:

$$13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1.$$

Notamos de lo anterior que la secuencia que comienza en 13 y termina en 1 contiene 10 términos. Aunque no está demostrado, se cree que al aplicar f repetidas veces sobre cualquier número natural, eventualmente se llegará a 1. Dentro de los números naturales menores que un millón: ¿a cuál le debo aplicar f la mayor cantidad de veces antes de llegar a 1?

f) [Problema 19] Suponiendo que el día 1 de Enero de 1900 fue lunes, y que los años biciestos ocurren todos los años divisibles por 4 excepto los años que son divisibles por 100 pero no por 400 (por ejemplo, 200, 700 y 1300). ¿Cuántos domingos ocurrieron el primer día de algún mes entre el primero de enero de 1901 y el 31 de diciembre de 2000?

g) [Problema 20] La suma de los dígitos de 10! es 27, pues $10! = 3628800$. Encuentra la suma de los dígitos de 100!.

h) [Problema 32] Decimos que un número de n dígitos ($n \leq 9$) es pandigital si usa todos los números entre 1 y n exactamente una vez. Por ejemplo, el número 15234 es pandigital, pues tiene 5 dígitos y menciona exactamente una vez cada dígito entre 1 y 5.

Si consideramos la multiplicación $39 \cdot 186 = 7254$, notamos que la concatenación de los factores y el producto es 391867254, que es un número pandigital.

Encuentra todas las multiplicaciones con la propiedad definida anteriormente, y calcula la suma de los resultados de dichas multiplicaciones.

i) [Problema 35] Decimos que el número 197 es un *primo circular* porque al *rotar* sus dígitos, obtenemos sólo números primos: 197, 719 y 971. Encuentra la multiplicación de todos los primos circulares menores a 50000.
Nota: Este ejercicio ha sido levemente modificado.

Instrucciones y entrega

Para realizar los ejercicios anteriores sólo puedes ocupar objetos numéricos, no están permitidos otros tipos de objetos como strings, listas, tuplas, etc. Además, no puedes importar módulos externos a lo que se menciona a continuación.

En las siguientes instrucciones, `nroalumno` representa tu número de alumno (si termina en j, se usa j minúscula). Deberás crear dos archivos. El primero, de nombre `modulo_nroalumno.py`, puede contener todas las funciones que quieras. Estas funciones pueden recibir una cantidad arbitraria de parámetros pero deben retornar un valor numérico. El segundo archivo debe contener nueve funciones llamadas `ejercicio_a`, `ejercicio_b`,..., `ejercicio_i`, que no reciban parámetros y que calculen y retornen lo pedido en el ejercicio correspondiente. El nombre de este archivo debe ser `nroalumno.py`. Además, este archivo debe importar al primer archivo como un módulo. Tu entrega deberá cumplir los siguientes requisitos:

- Ambos archivos han de seguir al pie de la letra las recomendaciones de la guía de estilo PEP8 para código Python que puedes encontrar en `http://legacy.python.org/dev/peps/pep-0008`. Considera que hay herramientas que ayudan a seguirlas. Una buena idea es instalar Sublime Text 3 y utilizar Package Control para instalar el paquete Anaconda.
- Cada una de las funciones `ejercicio_a`, `ejercicio_b`,..., `ejercicio_j` debe ejecutarse en menos de 30 segundos. Si usas heurísticas (como por ejemplo iterar sólo sobre números de la forma $6n + 1$ y $6n - 1$ para buscar primos), debes escribir un comentario en tu programa indicando por qué la heurística es válida.

Una vez que tengas los dos archivos descritos anteriormente, deberás comprimirlos en un solo archivo de nombre `nroalumno.zip` que tendrás que entregar en el buzón de SIDING que se creará para esta guía.

Entregar antes del 17 de septiembre a las 23:59:59.