



IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Actividad 5

Objetivos

- Aplicar técnicas de recursión y backtracking para resolver problemas complejos.

Entrega

- **Lenguaje a utilizar:** Python
- **Lugar:** Github
- **Hora:** 16:55
- **Desarrollo en parejas**

Introducción

Instrucciones

En esta actividad se desarrollarán cuatro problemas distintos, los cuales requieren distintos métodos y algoritmos de resolución. Los problemas están puestos en orden creciente de dificultad.

Problema 1: Warm up - Dificultad: Bastián (Lacayo)

Deben programar un algoritmo recursivo que calcule el máximo común divisor entre el factorial de dos números. El input serán dos números y el output debe ser el máximo común divisor entre el factorial de cada uno. Para este ejercicio **no se pueden** usar ciclos **for** ni **while**.

Ejemplo:

Input: 4 3

Output: 6

Consideremos que $4!$ es 24 y $3!$ es 6. El máximo común divisor entre ambos es 6, por lo que este es nuestro output.

Problema 2: Números perfectos - Dificultad: Laika¹ (Perro de Bastián)

Un número perfecto es un número natural que es igual a la suma de sus divisores propios positivos. Así, 6 es un número perfecto porque sus divisores propios son 1, 2 y 3; y $6 = 1 + 2 + 3$. Deberán hacer un programa que reciba como input un número y que entregue de output todos los números perfectos hasta ese número. En caso de no haber, retornar 0. Para este ejercicio **no se pueden** usar ciclos **for** ni **while**.

Ejemplo:

Input: 10

Output: 6

Opcional: Intentar mejorar el algoritmo guardando estados, es decir, en vez de calcular divisores constantemente, usar los ya calculados previamente para no volver a hacer el cálculo.

Problema 3: Laberinto - Dificultad: Hugo (Ayudante)

En este problema deberán resolver un laberinto. El input del problema es una lista de listas (matriz) con la representación del laberinto en donde las paredes se mostrarán como "X", los caminos se mostrarán como "0" y la salida estará marcada con una S. El laberinto siempre comenzará en la esquina superior izquierda (posición (0,0)). El output del problema debe ser una lista de tuplas en donde se muestre el camino a seguir para resolver el laberinto. Ejemplo:

Input: `[[0,X,0,X],[0,0,0,X],[0,X,0,X],[X,X,S,X]]`

Output: `[(0,0),(1,0),(1,1),(1,2),(2,2),(3,2)]`

Problema 4: Cryptoaritmética - Dificultad: Hans (Difícil)

El objetivo es asignar a cada letra un dígito de 0 a 9, para que la suma de dos palabras tenga sentido numérico. Las únicas reglas son que cada asignación de una misma letra debe tener el mismo dígito asignado, y un dígito no puede estar asignado a más de una letra. El input serán tres palabras, las tercera debe ser la suma de las dos primeras.

Ejemplo:

Input: SEND MORE FOOD

Output: 1034 7520 8554

Explicación:

$$\begin{array}{r} 1034 + 7520 = 8554 \\ \text{SEND} + \text{MORE} = \text{FOOD} \\ \text{S}=1, \text{E}=0, \text{N}=3, \text{D}=4, \text{M}=7, \text{O}=5, \text{R}=2, \text{F}=8. \end{array}$$

¹Laika demoró 5 minutos en hacer este ejercicio, Bastián 10

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.