



IIC2115 – Programación como Herramienta para la Ingeniería (I/2021)

## Taller 2b

### Objetivos

- Utilizar algoritmos y técnicas de programación, en conjunto con estructuras de datos, para resolver problemas.

### Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **T2b**.
- **Entrega:** lunes 19 de abril a las 16:50 hrs.
- **Formato de entrega:** archivos python notebook (**T2b.ipynb**) y python simple (**T2b.py**) con la solución a lo solicitado en el enunciado. Ambos archivos deben estar ubicados en la carpeta **T2b**. No se debe subir ningún otro archivo a la carpeta. En el archivo python notebook, utilice múltiples celdas de texto y código para facilitar la revisión de su programa. El archivo python simple debe ser exportado a partir del python notebook.
- **NO SE ADMITEN ENTREGAS POR CORREO U OTROS MEDIOS NI ENTREGAS FUERA DE PLAZO**
- **Entregas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

### Introducción

Con el fin de ejercitar el uso de técnicas de programación, en este taller deberá desarrollar 2 ejercicios.

## Ejercicio 1: profundidad de un árbol binario

Escriba un programa que calcule mediante recursión, la profundidad de un árbol binario, es decir, el camino más largo desde el nodo raíz hasta un nodo hoja. Empaque el algoritmo en una función que reciba como argumento una lista de tuplas de 3 elementos, que denotan el nodo padre, hijo izquierdo e hijo derecho, respectivamente. Cada nodo se identificará de manera única con un número natural. La ausencia de un hijo será representada a través de `None`, mientras que los nodos hoja no tendrán asociada una tupla con ellos como nodo padre. Puede asumir que la primera tupla de la lista representa al nodo raíz. La función debe retornar un número natural, que indique la profundidad del árbol. Un ejemplo de ejecución del algoritmo es el siguiente:

### Código

```
arbol = [(0,1,2), (1,3,None), (2,4,5), (4,None,6)]
profundidad = profundidad_arbol_binario(arbol)
print(max_long)
```

### Salida

3

## Ejercicio 2: grupos de estudio

Considere un conjunto de estudiantes, donde cada uno tiene asociado el puntaje que obtuvo en la última prueba rendida. Asumiendo que este puntaje corresponde a un número natural positivo, escriba un programa basado en *backtracking*, que dado el conjunto de alumnos y un número natural positivo  $K$ , entregue  $K$  grupos de estudio disjuntos, tales que la suma de los puntajes de los alumnos en cada grupo sea la misma. El retorno debe ser una lista de listas, donde cada sublista representa un grupo de estudio y contiene los índices de los alumnos que la componen. Un ejemplo de ejecución del algoritmo es el siguiente:

### Código

```
puntajes = [7,3,5,12,2,1,5,3,8,4,6,4]
K = 5
grupos = grupos_estudio(puntajes, K)
print(grupos)
```

### Salida

```
[[4,10,9], [8,11], [1,5,2,7], [3], [0,6]]
```

## Objetivo parcial de participación

Para verificar la participación durante la clase, debe terminar y entregar el ejercicio 1.

## 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.