



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

Taller 2a

Objetivos

- Resolver problemas de programación de manera eficiente y utilizar estructuras de datos avanzadas.

Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **T2a**.
- **Entrega:** lunes 13 de septiembre a las 16:50 hrs.
- **Formato de entrega:** archivos python notebook (**T2a.ipynb**) y python simple (**T2a.py**) con la solución a lo solicitado en el enunciado. Ambos archivos deben estar ubicados en la carpeta **T2a**. 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 .py simple debe ser exportado a partir del .ipynb.
- **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 continuar ejercitando el uso de estructuras de datos, en este taller deberá desarrollar un ejercicio de tres maneras distintas.

Ejercicio: el mayor del curso

En un colegio ubicado en El Tíbet, los estudiantes están obsesionados con la edad y deciden hacer una competencia para premiar a aquellos que mejor conocen la edad de sus compañeros. La competencia consiste en ser, al momento de ubicarse en la fila para entrar, mayor que todos los estudiantes que restan por ponerse en la fila. Dada la gran cantidad de alumnos del colegio, el director le pide que desarrolle un programa en Python, que dada las edades de todos los estudiantes que están en la fila (de izquierda a derecha), indique cuáles de estos son los ganadores de la competencia (sin importar el orden) a través de la posición en que se ubican estos en la fila (partiendo de 0).

Para resolver este problema, explore tres esquemas de solución:

- a) Fuerza bruta: solucione el problema sin utilizar estructuras de datos avanzadas, recorriendo los datos con un “doble `for`”.
- b) Stack: utilice esta estructura de datos para resolver el problema de manera más eficiente.
- c) Minimalista: solucione el problema con un único ciclo `for` y sin utilizar estructuras de datos.

Independiente el esquema de solución, un ejemplo de ejecución del programa debería verse de la siguiente manera:

Código

```
fila = [10, 4, 6, 3, 5]
ganadores = mayores_x(fila)
print(ganadores)
```

Salida

```
[0, 2, 4]
```

Donde `_x` en `mayores_x` indica el esquema de solución utilizado. Por ejemplo, para el esquema a) la función debiese tener por nombre `mayores_a`.

Para verificar el nivel de eficiencia de cada solución, una vez que haya completado las tres implementaciones, contraste la cantidad de comparaciones realizadas por todos esquemas de solución para un mismo valor de la entrada. Para esto, modifique las funciones de modo que retornen, además de la lista con la solución, la cantidad de comparaciones realizadas para encontrarla.

Objetivo parcial de participación

Para verificar la participación durante la clase, debe resolver el problema utilizando el esquema de solución

a) Fuerza bruta.

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.