



IIC2115 – Programación como Herramienta para la Ingeniería (I/2019)

Laboratorio 4 - Estructuras de datos

Objetivos

- Aplicar conocimientos de estructuras de datos en general, a través de la resolución de problemas específicos.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L03**.
- **Entrega parcial:** lunes 15 de abril a las 16:50 hrs.
- **Entrega final:** domingo 21 de abril a las 23:59 hrs.
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución de los distintos problemas ubicado en la carpeta **L04**. Suba además, en la misma carpeta, un archivo **README.md** con las instrucciones para ejecutar su tarea. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea.
- **Descuentos:** se descontará 0.5 pts. por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Entregas parciales subidas fuera de plazo no serán consideradas.**
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.

Introducción

Resolución de problemas

En este laboratorio deberán resolver dos problemas avanzados sobre estructuras de datos. Cada problema es independiente y se espera que sean resueltos utilizando los conocimientos del capítulo 2 del curso. No es necesario que utilicen **sólo** las estructuras de datos descritas en el material de clases, pueden utilizar cualquier estructura que facilite la resolución del ejercicio.

Problemas

I. Suma de cuadrados (1.0 pto.)

Dada una lista L de números naturales, encuentre todos los tríos $(L[i], L[j], L[k])$ con $i \neq j \neq k$, tales que $L[i]^2 + L[j]^2 = L[k]^2$. La solución entregada no puede estar basada en fuerza bruta, *i.e.*, no se puede solucionar el problema probando cada uno de los posibles tríos (i, j, k) .

Entrada:

[2,3,4,6,7,12,13,15,5,17,14,22]

Salida:

[[3,4,5],[5,12,13]]

II. Los Gremlins (2.0 ptos.)

Bastían se ha encontrado un Gremlin (de la película [Gremlins \(1984\)](#)), un ser muy felpudo y adorable que adopta como mascota. Averiguando sobre estos seres, Bastían encontró cuatro advertencias de sus cuidados:

- (a) Si tocan el agua se multiplican N veces. ($1 \leq N \leq 10^5$)
- (b) El propietario de la criatura no puede decidir cuanta comida darle.
- (c) Todos los días se les debe entregar la misma cantidad de alimento.
- (d) Con más de una criatura, si al final de un día, uno de sus Gremlin ha comido más que el Gremlin de la izquierda, entonces se convierte en demonio y se escapa.

Bastián no hizo caso de la primera regla y ahora tiene N de estas criaturas. Para no errar en las otras reglas, los pone en una hilera de jaulas para tenerlos numerados y saber las posiciones de estos.

Para cumplir con las reglas Bastián le pide a un amigo que le escriba lo que van a comer sus Gremlins diariamente. Con esa información quiere saber a partir de qué día ya no se transforman en demonios y así volver a casa (*i.e.* número de días en que no hay Gremlins con más alimento que el de su izquierda).

Por ejemplo, si el amigo le entrega la lista c de tamaño N donde cada elemento es la comida diaria de uno de los Gremlins, $c=[4,7,3,8,6]$. En el día **1** los Gremlins en posiciones 1 y 3 se convierten en demonios, quedando $c=[4,3,6]$. En el día **2** el Gremlin 2 se convierte en demonio, quedando $c=[4,3]$. Luego no se crean más demonios, por lo que la respuesta es “desde el día **3**”. La solución entregada no puede estar basada en fuerza bruta, recuerde que tiene un tiempo máximo de ejecución.

Entrada:

[6,5,8,4,7,10,9]

Salida:

3

III. Las fichas luminosas (3.0 ptos.)

Jaime tiene unas fichas luminosas que representan números enteros. Cada vez que se tocan cambian de color entre rojo y azul. A Jaime se le ocurre un juego que consiste en lo siguiente:

- poner n fichas en una hilera, $[r_1, r_2, \dots, r_n]$, inicialmente todas rojas.
- escoger dos posiciones i y j ($i < j$), y tocar todas las fichas entre la i -ésima y j -ésima ficha inclusives. Es decir, $[r_i, r_{i+1}, \dots, r_j]$ ahora serán azules.
- evaluar si el producto entre la i -ésima y j -ésima ficha (primera y última azul) es menor o igual a máximo valor de entre todas las fichas azules, es decir:

$$r_i \cdot r_j \leq \max(r_i, r_{i+1}, \dots, r_j) \quad (1)$$

Como Jaime es programador, decide analizar su juego y saber cuantos pares (i, j) cumplen su condición. Por lo que debe implementar una función que realice este análisis, es decir, que reciba una secuencia de los valores de las n fichas y retorne el número de pares que cumplen la condición de Jaime.

Entrada:

[1,1,2,4,2]

Salida:

8

Corrección

Para la corrección de este laboratorio, se revisarán dos aspectos, cada uno con igual valor en la nota (50%). El primero serán los contenidos y mecanismos utilizados para resolver cada uno de los problemas propuestos. De este modo, es importante que comente correctamente su código y que sea más sencilla la corrección. Recuerde que debe utilizar las estructuras de datos adecuadas a cada problema. Además, se evaluará el orden de su trabajo.

El segundo aspecto será la correctitud de los resultados y el tiempo de ejecución de las soluciones entregadas. Por cada problema se probarán distintos valores de entrada, y para cada uno de estos, el tiempo de respuesta no deberá ser superior a 1 segundo. Por lo tanto, para cada caso probado, si el resultado entregado por el algoritmo no es correcto, o si el tiempo de resolución supera **1 segundo**, la respuesta será considerada como incorrecta.

Avance parcial (final del la clase)

- Deberá entregar resuelto el problema 1.

Entrega final

- Entregar la solución de ambos problemas propuestos.

Política de Integridad Académica

“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.