



IIC2115 – Programación como Herramienta para la Ingeniería (I/2018)

Actividad 3

Objetivos

- Usar Estructuras de Datos para resolver problemas de forma eficiente.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** GitHub
- **Hora:** 16:55
- **Desarrollo individual o en parejas**

Introducción

En esta actividad, deberás resolver una serie de problemas (de dificultad incremental), usando de manera conveniente las estructuras de datos descritas en el material de clases. Todos los problemas son independientes entre sí, por lo que no hay necesidad de modelarlo todo junto. Se recomienda leer la actividad completa antes de comenzar a trabajar. Al terminar la clase (y no necesariamente la actividad), escriban un comentario al inicio de su código dando a conocer su opinión y/o su percepción de la actividad.

Recuerde que para el manejo de archivos existen los métodos `open`, `readline`, `readlines`, y `close`. Como referencia, puede ingresar a este [link](#).

Problema 1: Supermercado - Dificultad: Bastián (Lacayo)

En el supermercado *Colista*, se le pide ayuda para administrar los productos que están disponibles para los clientes, y las cajas en las que se paga por ellos. El supermercado tiene solamente 1 pasillo (por algo es el último en su rubro), en el cual se encuentran los productos organizados en estantes. El pasillo posee 3 estantes, y cada estante a su vez tiene 5 productos distintos, es decir, existe una lista (pasillo) de listas (estantes) de stacks o pilas (productos). Cuando un cliente desea comprar un producto, lo extrae de la fila en donde se encuentra ese producto sacando siempre el producto más visible, y lo guarda en su carro de compras. Una vez que el cliente ya está listo para pagar, se pone en la fila de la única caja en *Colista*, y espera para ser atendido (la fila funciona como una cola). Utilizando como base el archivo *productos.py*, modele el resto del sistema utilizando las estructuras de datos adecuadas, de manera de que se pueda realizar una simulación simple del sistema: 1 cliente elige 3 productos y los pasa por caja.

Problema 2: Inventario - Dificultad: Laika (Perro de Bastián)

Los ejecutivos del supermercado desean hacer un inventario sobre los tipos de productos que se venden actualmente. Cada producto pertenece a un departamento, y se desea obtener información sobre los departamentos. Para esto, se entrega un archivo con todos los productos en el supermercado (*productos.py*), y deberá agruparlos utilizando cada departamento como llave. A su vez cada departamento tendrá información sobre sus productos almacenados como datos inmutables (para que agentes maliciosos no modifiquen la información). Finalmente se deben imprimir en pantalla las siguientes estadísticas:

- Precio promedio por departamento.
- Stock total por departamento.

Problema 3: Organigrama - Dificultad: Felipe (Ayudante)

Usted ha sido contratado en una gran empresa, por lo que ingresa emocionadamente a la página web para confirmar su flamante nuevo puesto en el organigrama. Sin embargo, una mala mantención por parte del departamento de informática ha dejado la página ilegible. Con sus habilidades computinas, ha logrado extraer un diccionario que indica atributos relevantes de cada posición. Sin embargo, usted quiere ver esta información de manera jerárquica.

Dado el diccionario entregado, instancie y pueble un árbol que almacene el organigrama de la empresa. Luego de esto, deberá implementar un algoritmo de búsqueda para encontrar al reponedor “Juan Pérez” dentro del árbol. Por un tema de eficiencia, decide implementar el algoritmo IDS (Iterative Deepening Search¹), el que puede verse como hacer BFS sobre su árbol (de profundidad n) hasta una profundidad p . Inicialmente $p = 1$, y eventualmente $p = n$ si es que el resultado se encuentra en el último nivel de profundidad (o bien no se encuentra en el árbol).

Desafío Extra: Manejo de Bodega - Dificultad: Hans (Deidad)

Usted es el administrador de una bodega (su predecesor fue despedido por incompetencia). Entre otras cosas, debe organizar la llegada de los productos de sus proveedores. Para resolver el mal manejo de la atención a éstos, su jefe le ha encargado simular la situación para un período de 4 horas, que corresponde al funcionamiento actual.

La entrada a la bodega tiene 3 colas de espera para los furgones y camiones que van llegando. Cada 5 minutos, llega uno nuevo, que deberá integrarse a la cola con la menor cantidad de vehículos ya en espera (ante empate, podrá ser aleatorio). Cada 8 minutos, un vehículo de la cola con mayor cantidad de vehículos entra a la bodega, en donde hay 3 líneas de descarga de productos a los estantes. Sin embargo, debido a la mala administración actual, en todo momento hay sólo una línea activa a la que ingresan los vehículos desde las colas. Un vehículo sale de la bodega al ser descargado, lo que demora 10 minutos desde que comienza a ser atendido por el reponedor.

Cada hora hay cambio de turno, y el reponedor activo es reemplazado. Lamentablemente, estos reponedores son mañosos, y sólo trabajan en *su* línea. Esto significa que al ocurrir el cambio de turno, usted deberá mover todos los vehículos desde una línea a la otra, manteniendo el orden original². Claramente, solo se puede mover un vehículo si es el último en la línea (el más lejano a los estantes). Para poder cumplir su

¹https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search

²Puede asumir que el reordenamiento es instantáneo, el desafío es *planearlo*.

objetivo, puede usar todas las líneas como guste³. Recuerde usar inteligentemente las estructuras de datos que conoce para facilitar su trabajo.

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.

³Hint: Piense recursivamente.