



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

Laboratorio 3 - Estructuras de datos

Objetivos

- Aplicar los contenidos de estructuras de datos, 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 8 de abril a las 16:50 hrs.
- **Entrega final:** domingo 14 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 **L03**. 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 una serie de problemas cortos sobre estructuras de datos. Cada problema es independiente de los demás y se espera que sean resueltos utilizando los conocimientos del capítulo 2 del curso.

Para obtener el máximo de puntaje por problema, estos deben ser solucionados usando los tópicos indicados en cada uno de ellos. De lo contrario, sólo obtendrán una parte pequeña del puntaje.

Problemas

I. Dificultad básica (2.0 ptos.)

- a) **Camino máximo (Listas):** Considere dos listas de números enteros hexadecimales, que se encuentran ordenados de menor a mayor (más información sobre qué es un número hexadecimal [aquí](#)). Encuentre el camino de suma máxima compuesta por elementos de ambas listas, partiendo por el primer elemento de cualquiera de las dos, y sólo pudiendo cambiarse de lista cuando se utilice un número común entre ambas. Un ejemplo de ejecución del algoritmo es el siguiente:

Entrada:

```
['3', '6', '7', '8', '10', '12', '15', '18', '100']  
['1', '2', '3', '5', '7', '9', '10', '11', '15', '16', '18', '25', '50']
```

Salida:

```
['1', '2', '3', '6', '7', '9', '10', '12', '15', '16', '18', '100']
```

- b) **Sistema de registros (Diccionarios):** Se desea construir un sistema de registro de usuarios para una plataforma de correo electrónico. El sistema debe recibir una lista de nombres de usuario, y responder i) OK, en caso que el nombre de usuario no exista previamente, o ii) el nombre de usuario solicitado, concatenado con un número entero que indica la cantidad de usuario con el mismo nombre de usuario que existen en la base de datos de usuarios. Un ejemplo de ejecución del sistema es el siguiente:

Entrada:

```
['usuario','usuario','cliente','cliente','cliente']
```

Salida:

```
['OK','usuario1','OK','cliente1','cliente2']
```

II. Dificultad media (2.0 ptos.)

- a) **Secuencia de valor mínimo (Stacks):** Considere una secuencia compuesta sólo por los caracteres + y -. Cada una de las secuencias de este tipo codifica un número entero, tal que un carácter + indica que en la secuencia asociada se debe agregar un número entero mayor que el anterior, mientras que un carácter - indica que en la secuencia asociada se debe agregar un número entero menor que el anterior. Por ejemplo, la secuencia ++--++, puede decodificarse en los números 1232123, 5896478, 4565456, etc. Dado esto, para cada secuencia válida de caracteres + y -, encuentre el número entero de menor valor que no tenga dígitos repetidos. Un ejemplo de ejecución del algoritmo es el siguiente:

Entrada:

```
'++--++--'
```

Salida:

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

- b) **Las aldeas (Colas):** Considere un tablero de ($m \times n$) celdas. Cada letra **D** corresponde a celda de desierto, mientras que las **H** corresponden a celdas hogar. Una aldea esta compuesta por dos o más hogares conectados por aristas de celda. Construya una función capaz de recibir un tablero de este tipo y retornar el número de aldeas. Un ejemplo de ejecución del algoritmo es el siguiente:

Entrada:

```
[['D','H','D','D','D','D','D'],  
 ['D','D','D','D','D','H','H'],  
 ['D','H','D','D','H','H','H'],  
 ['D','H','H','D','H','D','D'],  
 ['D','D','D','D','D','D','D'],
```

```
['D','H','D','D','D','D','D'],  
['D','D','H','D','H','H','H'],  
['D','D','D','D','D','H','D']]
```

Salida:

3

III. Dificultad avanzada (2.0 ptos.)

Orden de una lista: Considere una lista de tamaño N , que contiene todos los números enteros en el intervalo $[1, N]$. Encuentre el orden inicial de esta lista, tal que el siguiente algoritmo se ejecute correctamente sobre la lista:

Para $n = [1...N]$

- (a) El primer elemento de la lista se mueve hasta la última posición. Este proceso se repite n veces.
- (b) El primer elemento de la lista es ahora el número n , que es removido de la lista, dejándola con tamaño $n - 1$.

Un ejemplo de ejecución del algoritmo es el siguiente:

Entrada:

4

Salida:

[2,1,4,3]

Hint: Considere el uso de algunas de las siguientes estructuras de datos: lista ligada, *segment tree*, *binary indexed tree*.

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 un problema.

Entrega final

- Entregar la solución de todos los 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.