



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

Actividad Práctica 2

Objetivos

- Evaluar la utilización adecuada de algoritmos y técnicas de programación, en conjunto con estructuras de datos, para resolver problemas.

Condiciones de desarrollo

- Evaluación estrictamente individual.
- No están permitidas las discusiones sobre la evaluación fuera de los canales oficiales (Discord, Zoom, sala de clases).
- Grupo de Whatsapp del curso debe permanecer cerrado durante todo el día, a partir del momento que se inicia la evaluación.

Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **A2**.
- **Entrega:** lunes 27 de septiembre a las 18:30 hrs.
- **Formato de entrega (ES IMPORTANTE EL NOMBRE DEL ARCHIVO):**
 - Archivo python notebook (**A2.ipynb**) con la solución de los problemas y (eventualmente) ejemplos de ejecución. Utilice múltiples celdas de texto y código para facilitar la revisión de su laboratorio.

- Archivo python (**A2.py**) con la solución de los problemas. Este archivo solo debe incluir la definición de las funciones, utilizando exactamente los mismos nombres y parámetros que se indican en el enunciado, y la importación de los módulos necesarios. Puede crear y utilizar nuevas funciones si lo cree necesario. No debe incluir en este archivo ejemplos de ejecución ni la ejecución de dichas funciones. **No deje instancias del método print() en el archivo .py.** Si desea mantener ejecuciones o prints en el archivo **.py** estos **DEBEN** estar dentro de un bloque main.
- Todos los archivos deben estar ubicados en la carpeta **A2**. No se debe subir ningún otro archivo a la carpeta.

- **NO SE ADMITEN 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 evaluar el uso de estructuras de datos y técnicas de programación, en esta actividad práctica deberá desarrollar 3 ejercicios. Para cada uno no solo es fundamental utilizar alguna(s) de las técnicas y/o estructuras cubiertas en el material del curso, sino que además hacerlo de manera adecuada.

Corrección

Todos los problemas tienen el mismo puntaje (**2,0 ptos.**) Para la corrección, se revisarán dos ítems por problema. El primero serán las estructuras y técnicas utilizadas para resolverlos. Estas deben ser apropiadas al problema y cumplir con los requisitos de complejidad indicados en el enunciado. De este modo, es importante que escriban código ordenado y que lo comenten explícitamente para que sea corregido de forma adecuada. Este ítem tendrá un valor de **1,5 ptos.** por problema.

El segundo ítem será la correctitud de los resultados, que será verificada mediante un revisor automático. Por cada problema se probarán distintos tamaños y valores de entrada, y para cada uno de estos se evaluará su correctitud. En particular, por cada tamaño de input probado, el resultado debe ser correcto para todas las instancias para obtener el puntaje. Un requisito fundamental para obtener puntaje en este ítem es respetar el formato de los nombres de funciones, y número y tipo de parámetros de entrada y valores de salida indicado en cada ejercicio. Este ítem tendrá un valor de **0,5 ptos.** por problema.

Ejercicio 1: bodegas ocupadas

Considere un sistema de bodegaje dispuesto de manera lineal, con k bodegas una bodega al lado de la otra, ubicado en un piso de un edificio. Con el fin facilitar el acceso a las bodegas, el piso posee entradas por ambos lados (izquierda y derecha). En base a esto, la administración decide asignar las bodegas vacías a los distintos clientes en base al lado por donde entran al piso. Por ejemplo, si un cliente entra por la izquierda, se le asigna la bodega vacía que está más a la izquierda del piso.

Para facilitar la gestión del sistema de bodegaje, la administración definió un formato de registro de eventos, que permite almacenar de forma eficiente el estado de las bodegas. Cada evento puede ser de tres tipos: llega cliente por la izquierda (I), llega cliente por la derecha (D), se vacía la bodega j (j). En base a esto, al finalizar el día, el estado final del sistema puede ser representado por una secuencia temporal de eventos. Asumiendo que i) cada día el sistema comienza con todas las bodegas vacías, ii) siempre que llegue un cliente habrá una bodega disponible y iii) siempre que se vaya un cliente de una bodega es porque él mismo la estaba usando, escriba un programa que dada la cantidad k de bodegas del sistema y una secuencia de n eventos, calcule el estado final del sistema de bodegas en la forma de una lista binaria (0 bodega desocupada, 1 bodega ocupada). La solución debe tener una complejidad no mayor a $\mathcal{O}(n)$.

Un ejemplo de ejecución del programa es el siguiente:

Código

```
k = 10
eventos = "I0I0IIDD9"
num_bodegas = bodegas_vacias(k, eventos)
print(num_bodegas)
```

Salida

```
[1,1,0,0,0,0,0,0,1,0]
```

Ejercicio 2: diferencias entre índices y valores

Dada una lista de n números enteros de tamaño arbitrario, escriba un programa que encuentre la cantidad de pares (i, j) de la lista, tales que $i < j$ y $a_j - a_i = j - i$. La solución debe tener una complejidad no mayor a $\mathcal{O}(n)$.

Un ejemplo de ejecución del algoritmo es el siguiente:

Código

```
lista = [3, 5, 1, 4, 6, 6]
num_pares = pares_misma_diferencia(lista)
print(num_pares)
```

Salida

1

Ejercicio 3: epifanía

En un curso universitario, sus n estudiantes discuten en sus respectivos grupos de estudio, qué ejercicios son los más adecuados para estudiar para la siguiente prueba. Estos grupos de estudio no son disjuntos, por lo que un estudiante puede estar en más de un grupo. En un momento dado, un estudiante no identificado tiene una epifanía, donde se le revela el contenido completo de la prueba.

Considerando esta situación, escriba un programa que dada la composición de los k grupos de estudio, encuentre aquel estudiante que maximizaría la difusión de los ejercicios en el curso, si este hubiese tenido la epifanía. La solución debe tener una complejidad no mayor a $\mathcal{O}(nk^2)$.

Un ejemplo de ejecución del algoritmo es el siguiente:

Código

```
grupos = [[1,4,3],[0,1],[0],[5,6],[2]]
max_difusion = difusion_epifania(grupos)
print(max_difusion)
```

Salida

0

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.