



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

Laboratorio 5 - Técnicas y Algoritmos

Objetivos

- Aplicar conocimientos de técnicas y algoritmos, 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 **L05**.
- **Entrega parcial:** No habrá entrega parcial
- **Entrega final:** domingo 28 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 **L05**. 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.
- **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

Resuelva los siguientes problemas con la ayuda de los algoritmos presentados en el capítulo 3. Usted deberá entregar tres funciones o métodos, uno por problema. Cada función debe solicitar los datos de entrada (como parámetros) y retornar la información solicitada. Junto a cada problema dispondrá de una indicación del tópico que debe utilizar para resolverlo.

Problemas

I. Dificultad básica (1.5 ptos.) Batalla de puntos (Recursión): Suponga que va a competir en una batalla de videojuegos. En ella puede realizar ataques que le suman 1 ó 2 puntos. ¿Cuántas maneras diferentes existen de atacar para juntar N puntos?. Por ejemplo, si desea juntar $N = 3$ puntos, existen tres maneras de atacar: atacar tres veces por 1 punto, atacar por 2 puntos y luego por 1 punto o atacar por 1 punto y luego por 2 puntos. Para un cantidad total de puntos N , se pide retornar la cantidad de maneras de atacar diferentes.

Entrada:

4

Salida:

5

II. Dificultad media (2.5 ptos.) Grupos de estudio (Backtracking): Considere un conjunto de N estudiantes, donde cada uno tiene asociado el puntaje que obtuvo en la última prueba rendida. Asumiendo que este puntaje corresponde a un número natural positivo, escriba un programa que dado el conjunto de alumnos y un número natural positivo K , entregue K grupos de estudio disjuntos, tales que la suma de los puntajes de los alumnos en cada grupo sea la misma.

Entrada:

5

[7,3,5,12,2,1,5,3,8,4,6,4]

Salida:

$[[2,6,4], [8,4], [3,1,5,3], [12], [7,5]]$

III. Dificultad avanzada (3.0 ptos.)

Viajeros frecuentes (Ordenamiento): La aerolínea BastiAir posee un algoritmo para detectar viajeros frecuentes y ofrecerles nueva publicidad. Este se basa en que si la cantidad de viajes realizados por un cliente en un año en particular es mayor o igual al doble de la mediana de viajes realizados por el cliente en un número específico de a años previos, la aerolínea enviará publicidad a ese cliente. La empresa no enviará ningún tipo de publicidad hasta que recopile al menos la cantidad de datos para realizar su algoritmo, es decir, cuando tenga datos de al menos a años previos.

Dado el “número específico de años a para el cálculo de mediana” y “la cantidad de viajes anuales para un periodo de n años” ($a < n$), encuentre y retorne la cantidad de veces que un cliente recibe publicidad dentro del periodo de n años.

Por ejemplo, $a = 3$ y $viajes = [10, 20, 30, 40, 50]$. Para los primeros 3 años la empresa solo ha recopilado la cantidad de información mínima. En el cuarto año, los viajes previos son $[10, 20, 30]$. La mediana es 20 y los viajes de ese año son 40. Dado que $40 \geq 2 \cdot 20$, recibirá publicidad ese año. Al quinto año viaja 50 veces y los viajes previos son $[20, 30, 40]$. Como $50 \not\geq 2 \cdot 30$, no será notificado el quinto año. Por lo tanto, para el período será notificado sólo una vez.

Entrada:

5

$[2, 3, 4, 2, 3, 6, 8, 4, 5]$

Salida:

2

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.

Entrega final

- Entregar la solución de los tres 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.