



Semana 1: Algoritmos

Programación

2018 - 2019

Grado en Ingeniería Informática

Instrucciones generales

Los ejercicios se realizarán **semanalmente**.

La puntuación máxima total obtenible por la entrega de estos ejercicios es de 0,5 puntos. Para obtener puntuación en este apartado es necesario haber entregado al menos el 80% de los ejercicios.

En la clase de prácticas posterior a la entrega de cada ejercicio semanal, se solicitará a algún estudiante que explique **oralmente** la solución a uno de los ejercicios entregados. La exposición oral tendrá una puntuación total máxima de 0,5 puntos.

Salvo en esta primera semana, en que la entrega será individual, las prácticas deberán realizarse **obligatoriamente en grupos de dos estudiantes** (hablar con el profesor de prácticas en caso de ser impares o de tener problemas para encontrar pareja). Si por alguna razón justificada quieres cambiar de pareja de prácticas habla con tu profesor.

Se puede entregar cualquier práctica semanal aunque no se haya entregado la anterior

Aunque el peso específico de las prácticas semanales en la nota global de la asignatura puede parecer pequeño, la realización de estos ejercicios semanales es fundamental para la comprensión de la asignatura y para obtener el conocimiento necesario para aprobar el examen final. Estos ejercicios semanales también constituyen la base para la realización del proyecto final, por lo que es muy recomendable seguir la asignatura desde el principio realizando todas las prácticas asignadas.

Introducción a los algoritmos

Un algoritmo es una secuencia de pasos que se utiliza para resolver un problema. Un ejemplo es el algoritmo de cambio de la rueda de un coche que vimos en la clase de teoría. O el siguiente para comprar un billete de metro en una máquina:

1. Ir a la máquina.
2. Elegir el tipo de billete.
3. Pagar.
4. Entrar al metro.

Este es el tipo más sencillo de algoritmos, los secuenciales, en los que todos los pasos se ejecutan uno detrás de otro. Pero salvo para tareas triviales los algoritmos secuenciales no son muy útiles y es necesario añadir condiciones y repeticiones (bucles).

Un ejemplo de un algoritmo que incluye una condición sería el siguiente, que describe de forma un poco más compleja la compra de un billete de metro:

1. Ir a la máquina.
2. Elegir el tipo de billete.
3. Si voy a pagar en metálico ir al paso 4, si no ir al paso 6.
4. Pagar con monedas o billetes.
5. Ir al último paso
6. Pagar con tarjeta.
7. Entrar al metro.

En este caso, el paso 3 determina si el siguiente paso a ejecutarse es el 4 o el 6. Por lo tanto el paso 3 es una condición que utilizo para definir cómo continuo ejecutando mi algoritmo.

Un ejemplo de algoritmo con repetición, sería el siguiente, que extiende el anterior.

1. Ir a la máquina.
2. Elegir el tipo de billete.
3. Si voy a pagar en metálico ir al paso 4, si no ir al paso 8.
4. Pagar con monedas o billetes.
5. Si no hay vuelta que coger ir al último paso
6. Recoger una moneda de la vuelta
7. Ir al paso 5
8. Pagar con tarjeta.
9. Entrar al metro.

En este caso hay un bucle entre los pasos 5 y 7, en los que se repiten varios pasos mientras se cumple una condición.

Nota: en este caso habría un algoritmo mejor, que es el de recoger todas las monedas a la vez, en cuyo caso no haría falta el bucle.

Ejercicios

Ejercicio 1. Describir el algoritmo para hacer una tortilla de patatas.

Ejercicio 2. Describir el mismo algoritmo pero teniendo en cuenta que hay gente que no le echa cebolla.

Ejercicio 3. Describir el mismo algoritmo pero teniendo en cuenta que hay que cascar 5 huevos y que hay que hacerlo uno detrás de otro.

Ejercicio 4. Describir el algoritmo que debe utilizar un cajero automático para entregar dinero.

Ejercicio 5. Extender el algoritmo anterior para que el cajero compruebe si hay saldo suficiente en la cuenta antes de dar el dinero y si no lo hay no lo dé.

Ejercicio 6. Extender el algoritmo anterior para que si no hay saldo suficiente, el cajero pida al usuario que introduzca una nueva cantidad (lo debe hacer mientras el usuario no introduzca una cantidad correcta)

Ejercicio 7. Crear un algoritmo que compruebe si un número A es múltiplo de otro número B, usando solamente sumas, condicionales (comparaciones) y bucles.

Normas de entrega

Los ejercicios se deben **subir a Aula Global** hasta las 8:00 del Lunes 18 de Septiembre de 2018. Se deberá subir un un fichero comprimido **zip** que contenga los 7 archivos en formato **PDF**. El nombre del fichero será “s1-iniciales-del-alumno.zip” (por ejemplo Lucía Pérez Gómez subirá un fichero s1-lpg.zip). Se debe usar el enlace de entrega del grupo de laboratorio al que se pertenezca.