



UNIVERSIDAD  
DE GRANADA

*Este documento está protegido por la Ley de Propiedad Intelectual ([Real Decreto Ley 1/1996 de 12 de abril](#)).  
Queda expresamente prohibido su uso o distribución sin autorización del autor.*

# Algorítmica

## 2º Grado en Ingeniería Informática

### Guión de prácticas

#### Algoritmos Greedy

1. Objetivo.....	2
2. Descripción del problema.....	2
3. Tareas a realizar.....	2
4. Entrega de la práctica.....	3

© Prof. Manuel Pegalajar Cuéllar  
Dpto. Ciencias de la Computación e I. A.  
Universidad de Granada



DECSAI

**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

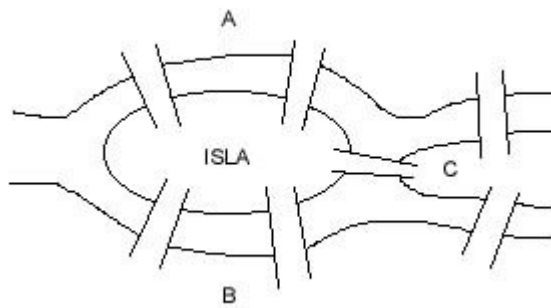
# Algoritmos Greedy

## 1. Objetivo

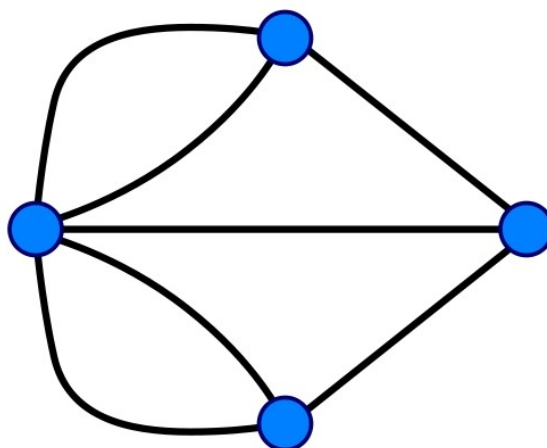
El objetivo de la práctica consiste en que el alumno sea capaz de analizar un problema, comprobar si puede ser resoluble mediante la técnica de diseño de algoritmos voraces, y aplicarla. Para ello, se expone un problema que deberá ser resuelto por el estudiante.

## 2. Descripción del problema

Durante el siglo XVIII, la ciudad de Königsberg en Prusia Oriental estaba dividida en cuatro zonas por el río Pregel. Había un total de 7 puentes que comunicaban estas regiones, tal y como se muestra en la siguiente figura:



Los habitantes de la ciudad realizaban paseos dominicales tratando de encontrar una forma de caminar por la ciudad, cruzando cada puente una sola vez, y regresando al lugar de partida. Para resolver este problema, en 1736, Leonard Euler representó las cuatro zonas como cuatro puntos, y los puentes como aristas que unen los puntos, tal y como se muestra en la siguiente figura:



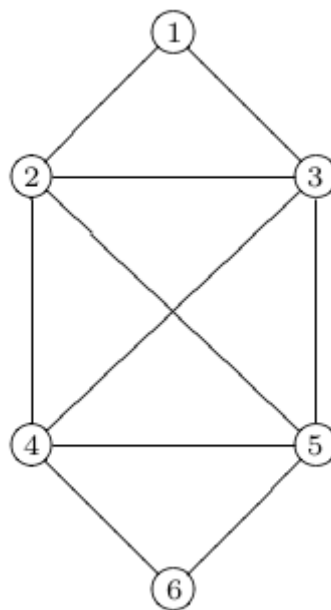
Encontrar el paseo a través de los puentes de Königsberg era equivalente a encontrar un circuito en el grafo anterior de modo que se pase una única vez por cada arista, también conocido como camino de Euler. En el caso de Königsberg, este camino es imposible de

encontrar.

Euler definió un tipo particular de grafo, denominado grafo de Euler, como aquel tipo de grafo que contiene un circuito de Euler. La definición formal de circuito de Euler es:

Sea  $G$  un grafo conexo. Un camino de Euler sobre el grafo es un camino que no repite aristas y en el que aparecen todas las aristas del grafo. Un circuito de Euler es un ciclo (comienza y termina en el mismo nodo/vértice) y que, además, es un camino de Euler.

Como ejemplo, el grafo de la siguiente figura es un grafo de Euler, donde el recorrido por los nodos (1, 2, 4, 6, 5, 3, 2, 5, 4, 3, 1) proporciona un circuito de Euler.



Se cumple que un grafo no dirigido conexo es también un grafo de Euler si, y sólo si, todos los nodos del grafo tienen grado par; es decir, si en todos los nodos del grafo inciden un número par de aristas. Existe un algoritmo (algoritmo de Fleury) que permite encontrar un camino de Euler sobre un grafo de Euler, cuya descripción de funcionamiento es la siguiente:

1. Se parte de un nodo dado  $v$  del grafo  $G$ .
2. Si  $G$  contiene sólo un nodo  $v$ , el algoritmo termina.
3. Si hay una única arista  $a$  que incide en  $v$ , entonces llamamos  $w$  al otro vértice que conecta la arista  $a$ , y la quitamos del grafo. Vamos después al paso 5. En otro caso, seguimos en el paso 4.
4. Como hay más de un lado que incide en  $v$ , elegimos uno de estos (lo llamamos  $w$ ) de modo que al quitarlo del grafo  $G$ , el grafo siga siendo conexo. Cogemos la arista que une  $v$  con  $w$  y la quitamos del grafo.
5. Cambiamos el nodo  $v$  por el nodo  $w$  y volvemos al paso 3 hasta que terminemos de hacer el circuito de Euler.

### 3. Tareas a realizar

Se pide:

1. Formalizar, si es posible, la descripción de funcionamiento del método anterior como un algoritmo Greedy. Para ello:
  1. Compruebe si se puede resolver mediante Greedy.
  2. Diseñe las componentes greedy del algoritmo.
  3. Adapte la plantilla de diseño Greedy a las componentes propuestas.
2. Implemente el algoritmo en una función C/C++.
3. Proponga dos ejemplos de grafos de Euler que se puedan leer desde fichero, y ejecute el programa implementado con estos dos ejemplos.
4. Calcule la eficiencia en el caso peor del algoritmo.

### 4. Entrega de la práctica

Se deberá entregar un **fichero ZIP** conteniendo:

- Uno o varios ficheros .cpp con el código fuente desarrollado (y si se considera necesario, un fichero makefile). Incluya los ficheros de texto con la definición de los grafos de ejemplo.
- Una memoria de prácticas en **PDF**, describiendo las soluciones a cada uno de los apartados. La memoria deberá incluir un apartado en el que se indique cómo compilar los algoritmos y cómo ejecutarlos para algún ejemplo de prueba.

La práctica deberá ser entregada por PRADO, en la fecha y hora límite explicada en clase por el profesor. No se aceptarán, bajo ningún concepto, prácticas entregadas con posterioridad a la fecha límite indicada. La entrega de PRADO permanecerá abierta con, al menos, una semana de antelación antes de la fecha límite, por lo que todo alumno tendrá tiempo suficiente para entregarla.

La práctica contribuirá con 2 puntos sobre 10, ponderado al total de la puntuación de prácticas expuesto en la guía docente de la asignatura.

El profesor, en clase de prácticas, podrá realizar auditorías de las prácticas a discreción, con el fin de asegurar de que los estudiantes alcanzan las competencias deseadas. Por este motivo, una vez finalizada la entrega de prácticas por PRADO, es recomendable repasar los ejercicios entregados para poder responder a las preguntas del profesor, llegado el caso de su defensa. La no superación de la defensa de prácticas supondrá una calificación de 0 en esta práctica. La superación de la defensa supondrá mantener la calificación obtenida.