

# Algorítmica - Practica 3: Algoritmos Voraces

*A. Herrera, A. Moya, I. Sevillano, J.L. Suarez*

*30 de abril de 2015*

## Contents

<b>1</b>	<b>Organización de la práctica</b>	<b>2</b>
<b>2</b>	<b>Problema 4</b>	<b>3</b>
2.1	Enunciado del problema . . . . .	3
2.2	Solución teórica . . . . .	3
<b>3</b>	<b>Problema 4</b>	<b>4</b>
3.1	Enunciado del problema . . . . .	4

# 1 Organización de la práctica

La práctica 3 trata sobre el desarrollo de algoritmos greedy que consigan la solución óptima de los problemas propuestos o actúen como heurística sobre los mismos. Uno de los problemas a estudiar es el viajante de comercio, ampliamente conocido en el ámbito de la inteligencia artificial y teoría de algoritmos. Es un problema NP completo pero que será abordable gracias al uso de algoritmos greedy polinomiales que no proporcionarán la solución óptima pero sí una lo suficientemente buena para nuestros objetivos.

Nuestro grupo debe resolver el problema 4 y el viajante de comercio. Hemos abordado también el problema opcional (número 5) opteniendo el algoritmo greedy óptimo.

Para cada problema se sigue la siguiente estructura:

- Enunciado del problema
- Resolución teórica del problema (con una subsección por algoritmo)
- Análisis empírico. Análisis de la eficiencia híbrida

En este último apartado se proporcionan gráficas con los resultados de los algoritmos y un análisis de la eficiencia híbrida para los mismos.

Los algoritmos se han ejecutado sobre un ordenador con las siguientes características:

- **Marca:** Toshiba
- **RAM:** 8 GB
- **Procesador:** Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

El código, los resultados de las ejecuciones, las gráficas y los pdf asociados se pueden encontrar en [GitHub](#).

## 2 Problema 4

### 2.1 Enunciado del problema

Consideremos un grafo no dirigido  $G = (V, E)$ . Un conjunto  $U$  se dice que es un recubrimiento de  $G$  si  $U \subseteq V$  y cada arista en  $E$  incide en, al menos, un vértice o nodo de  $U$ , es decir,

$$\forall (x, y) \in E : x \in U \text{ o } y \in U$$

Un conjunto de nodos es un recubrimiento minimal de  $G$  si es un recubrimiento con el menor número posible de nodos.

- Diseñar un algoritmo greedy para intentar obtener un recubrimiento minimal de  $G$ . Demostrar que el algoritmo es correcto, o dar un contraejemplo.
- Diseñar un algoritmo greedy que obtenga un recubrimiento minimal para el caso particular de grafos que sean árboles.
- Opcionalmente, realizar un estudio experimental de las diferencias entre los dos algoritmos anteriores cuando ambos se aplican a árboles.

### 2.2 Solución teórica

#### 2.2.1 Solución para un grafo arbitrario.

En primer lugar, nótese que si tomamos  $U = G$  tenemos un recubrimiento. Este será el peor recubrimiento posible pues utiliza todos los nodos del grafo. Queremos obtener un mejor recubrimiento. Para ello, construyamos uno desde 0:

**Algoritmo 1.** Un primer intento de algoritmo aleatorio.

1.  $U = \emptyset$
2. Para cada arista  $(x, y) \in E$  tomamos como nodo  $x$  o  $y$  aleatoriamente y lo añadimos a  $U$

Este algoritmo voraz aleatorio obtiene un recubrimiento del grafo pues para cada arista hay efectivamente un nodo en  $U$  sobre el que esta incide. El tiempo de ejecución es lineal sobre el número de aristas,  $\theta(|E|)$ . Sin embargo, es claro que las soluciones obtenidas serán manifiestamente mejorables. Podemos mantener este tipo de construcción pero librarnos parcialmente de la aleatoriedad introduciendo una elección voraz al algoritmo:

**Algoritmo 2.** Mejora del Algoritmo 1 realizando una elección voraz.

1.  $U = \emptyset$
2. Para cada arista  $(x, y) \in E$  tomamos un nodo  $v$  y lo añadimos a  $U$  donde  $v$  es:
  - $x$  si  $x \in U$ .
  - $y$  si  $y \in U$ .
  - Uno de los dos, elegido aleatoriamente, si  $x, y \notin U$ .

En efecto, si algún nodo sobre el que incide la arista ya está en  $U$  no tenemos por qué añadir uno nuevo.

Sin embargo, este algoritmo lineal y aleatorio no es óptimo. Para obtener un algoritmo voraz óptimo necesitamos construir la solución desde otra perspectiva.

En lugar de añadir un nodo por arista elegido bajo cierto criterio voraz busquemos otra forma de construir la solución. Consideramos para cada nodo su grado. El grado de un nodo es el número de aristas que inciden

en él. Es fácil ver que en la solución óptima debería estar el nodo con mayor grado ya que así no tenemos que preocuparnos de añadir un nodo para las aristas que inciden sobre él. Podemos abstraer este criterio como una función objetivo para nuestro algoritmo greedy, en cada iteración añadimos a  $U$  el nodo con mayor grado hasta que toda arista incida sobre algún nodo de  $U$ . Sin embargo, puede darse el caso de que estemos introduciendo un nodo innecesario pues las aristas que inciden en él ya inciden sobre algún otro nodo de  $U$ . Decidimos ante esta situación realizar la siguiente operación: cada vez que un nodo se añade a  $U$  se elimina el nodo de  $V$  y se eliminan de  $E$  las aristas que inciden sobre él. Posteriormente recalculamos el grado de cada nodo para este nuevo grafo. Nuestro algoritmo quedaría de la siguiente forma:

**Algoritmo 3.** Algoritmo óptimo.

```
U = []
while not E.isEmpty():
    v = V.nodeMaximumDegree()
    U.add(v)
    for edge in E:
        U.delete(edge) if edge[0] == v or edge[1] == v
```

Veamos que efectivamente este algoritmo obtiene la solución óptima.

## 3 Problema 4

### 3.1 Enunciado del problema

Un electricista necesita hacer  $n$  reparaciones urgentes, y sabe de antemano el tiempo que le va a llevar cada una de ellas: en la tarea  $i$ -ésima tardará  $t_i$  minutos. Como en su empresa le pagan dependiendo de la satisfacción del cliente y esta es inversamente proporcional al tiempo que tardan en atenderles, necesita decidir el orden en el que atenderá los avisos para minimizar el tiempo medio de atención de los clientes (desde el inicio hasta que su reparación es efectuada).

- Diseñar un algoritmo greedy para resolver esta tarea. Demostrar que el algoritmo obtiene la solución óptima.
- Modificar el algoritmo anterior para el caso de una empresa en la que se disponga de los servicios de más de un electricista.