

# 7529. Teoría de Algoritmos I

## Trabajo Práctico 2

Ferreyra, Oscar, *Padrón Nro. 89563*  
fferreyra38@gmail.com

Martin, Débora, *Padrón Nro. 90934*  
demartin@fi.uba.ar

Eisner, Ariel, *Padrón Nro. 90697*  
aeeisnerg@gmail.com

2do. Cuatrimestre de 2016

# Índice

<b>1. Clases de complejidad</b>	<b>2</b>
<b>2. Algoritmos de aproximación</b>	<b>2</b>
2.1. El problema de la mochila . . . . .	2
2.2. El problema del viajante de comercio . . . . .	2

## 1. Clases de complejidad

## 2. Algoritmos de aproximación

### 2.1. El problema de la mochila

### 2.2. El problema del viajante de comercio

Para implementar la solución aproximada a este problema se utilizó el algoritmo de Prim para calcular el árbol de tendido mínimo. La complejidad de dicho algoritmo depende de las operaciones de una cola de prioridad que contiene los vertices, de las iteraciones que se realizan sobre todas las aristas de cada vértice. Se tomará  $|V|$  como la cantidad de vértices del grafo y  $|A|$  como la cantidad de aristas. Las operaciones más importantes de la cola de prioridad son  $O(\log |V|)$  cada una, excepto el contains que es  $O(|V|)$ . Por otro lado el ciclo principal es  $O(|V|^2 * |A|)$  porque recorre para cada vertice, sus aristas, y verifica si su arista adyacente está en la cola de prioridad. El mencionado sería el peor caso, pero a medida que va avanzando la ejecución, la cola se va vaciando y el contains pasa a ser insignificante. Por último, se agregan todos los vertices y las aristas pertenecientes al árbol en un nuevo grafo. Esta operación es  $O(|A| \log |A|)$ , en el peor caso que sería que todas las aristas del grafo original debieran ser agregadas al nuevo grafo. Pero en la mayoría de los casos será mucho menor y dependerá de la cantidad de aristas del árbol.

Además del algoritmo de Prim, luego se busca el preorder del árbol de tendido mínimo. Este recorre todos los vértices del grafo por lo que es  $O(|V|)$ .

Los tiempos de ejecución fueron llamativamente inferiores a los obtenidos por programación dinámica, aunque las soluciones obtenidas no resultaron siempre iguales a las anteriores. A continuación se muestran dichos tiempos de ejecución. Es de notar que en esta ocasión sí fue posible realizar la corrida para grafos con mayor cantidad de vertices.

Cantidad de ciudades	Tiempo de ejecución (segundos)
15	$4,86 * 10^{-3}$
17	$5,74 * 10^{-3}$
19	$8,82 * 10^{-3}$
21	$6,35 * 10^{-3}$
23	$6,15 * 10^{-3}$
48	$13,4 * 10^{-3}$

Cuadro 1: Tiempo en función de la cantidad de ciudades para la aproximación por el árbol de tendido mínimo