



**UNIVERSIDAD
DE GRANADA**

**E.T.S. DE INGENIERÍAS INFORMÁTICA y DE
TELECOMUNICACIÓN**

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

Doble Grado en Ingeniería Informática y Matemáticas

Curso 2022-2023

Algorítmica

Guión de Prácticas

Práctica 3: Algoritmos voraces

Índice

1. Enunciado de los problemas	1
1.1. Servicio de <i>catering</i>	2
1.2. Codificación textual de imágenes	2
1.3. <i>Jukebox</i> de vinilos	3
1.4. Entrenadores personales y clientes	3
1.5. Grafos parciales con aristas negativas	4
1.6. El viajante de comercio (PVC)	4
2. Trabajo a realizar	4
2.1. Tareas	4
2.2. Memoria	5
3. Evaluación de la práctica	6

Objetivo

El objetivo de esta práctica es que el estudiante comprenda y asimile el funcionamiento de la técnica de diseño de algoritmos voraces. Esta práctica será realizada por grupos de alumnos. A cada grupo se le asignará un problema para el que tendrá que diseñar un algoritmo según las indicaciones incluidas en este guión. Además, tendrá que abordar el diseño de heurísticas, algoritmos aproximativos, basados en la filosofía de los métodos voraces para encontrar soluciones a un problema muy complejo, para el que no se conocen algoritmos eficientes.

En la sección que sigue se muestran los enunciados de los problemas a considerar.

1. Enunciado de los problemas

En esta sección se incluyen los enunciados de los problemas que tienen que resolver cada grupo de alumnos. Los problemas considerados son los cinco primeros que se describen esta sección. Para todos ellos hay que diseñar un algoritmo que dé una solución válida, es decir, que cumpla los requisitos y

que dé el óptimo cuando sea el caso. Sin embargo, el último problema, denominado “Viajante de comercio” es un problema especialmente complejo¹ y se sospecha que no existen algoritmos que den soluciones optimales para todos los casos en un tiempo “aceptable”. En ese caso, como *solución aceptable* se diseñan algoritmos que dan soluciones aproximadas, heurísticas. Este último problema habrá de ser abordado por todos los grupos de trabajo, además, del problema concreto que se les asigne.

1.1. Servicio de *catering*

Una empresa de *catering* debe servir a los n comensales de un banquete para lo que dispone de un total de c camareros, con $c < n$. Los comensales se encuentran sentados en mesas individuales, distribuidas por el local. Cada camarero sólo puede atender a un comensal a la vez. Cuando todos los camareros están ocupados atendiendo a comensales, el resto de comensales deberán esperar a que alguno de los camareros quede libre. Dado que un camarero puede llevar solamente un producto a la vez, el tiempo de espera de cada comensal dependerá de: 1) lo lejos que esté su mesa de la salida de cocina (desplazamientos) y 2) de los comensales que el camarero que le sirve la comida haya tenido que atender con anterioridad. Claramente, cuanto mayor es el tiempo de espera de un comensal mayor será el grado de insatisfacción con el servicio.

Se pide diseñar un algoritmo greedy que obtenga una asignación de comensales a camareros de forma que el servicio se haga con el menor tiempo medio de espera de los comensales posible.

1.2. Codificación textual de imágenes

Una imagen se almacena como una matriz bidimensional ($f \times c$) de píxeles. Cada pixel es codificado como un valor entero (`int`, representado con 64 bits). El sistema de captación utiliza una representación *raw*, es decir, se almacena expresamente cada píxel, utilizando un espacio total de $f \times c \times 8$ bytes. Sin embargo, hay situaciones en las que la distribución de colores de la imagen es bastante sesgada y puede ser conveniente utilizar una codificación distinta para los píxeles. De esta forma, podemos asignar a cada color de pixel un código con distinta longitud. Por simplicidad consideraremos que el código está formado únicamente por caracteres numéricos hexadecimales, es decir, $\{0, 1, \dots, 9, A, B, C, D, E\}$. Cada uno de estos símbolos se almacena usando un *nibble*, es decir, 4 bits. Por tanto, en cada byte se pueden

¹De la categoría denominada NP-duros.

almacenar 2 caracteres hexadecimales. Ejemplos de códigos a asignar serían $0, 0A, 12C, \dots$.

Se pide diseñar un algoritmo para, dada una imagen, encontrar una codificación de los colores presentes en la misma, tal que la representación basada en esta codificación sea lo más corta (en tamaño) posible.

1.3. *Jukebox* de vinilos

Para animar la estancia de los clientes de un centro comercial se ha instalado un reproductor de música, *jukebox*, en la que los clientes, a cambio de 10 céntimos pueden seleccionar una canción de entre los clásicos más populares para que se reproduzca. El encanto de este dispositivo es que la música se haya guardada en vinilos, no en soporte digital. El sistema tiene un amplio repertorio de música, pero organizado en dos grupos. Hay un primer grupo de vinilos cuya reproducción es inmediata. Es decir, en cuanto se selecciona la canción se puede reproducir. Sin embargo, el espacio disponible para este grupo de vinilos es limitado. El resto de vinilos están en un espacio adjunto. Si la canción que selecciona el cliente no está entre los vinilos disponibles para reproducción inmediata, se puede reproducir, pero antes hay que reemplazar uno de los vinilos en el grupo de disponibilidad inmediata por el vinilo que incluye la canción seleccionada.

Supongamos que el número total de canciones en la caja de música es c , y que el número máximo de canciones que se pueden incluir en el grupo de reproducción inmediata es k . Supongamos que de partida no hay ningún vinilo en este grupo de reproducción inmediata y que se irá rellenando con las peticiones de los clientes. Por simplicidad, asumamos que cada vinilo incluye sólo una canción.

Se pide diseñar un algoritmo para, dada una secuencia de peticiones de canciones, minimizar el número de reemplazos de vinilos dentro del grupo de reproducción inmediata.

1.4. Entrenadores personales y clientes

Un centro deportivo tiene en plantilla n entrenadores personales. Son profesionales de alta cualificación que durante un periodo se dedican en exclusiva a asesorar a un único cliente. El centro atiende a n clientes. Cada entrenador tiene una tarifa, $t_i, i = 1, \dots, n$, especificada como un ratio normalizado, es decir, $t_i \in [0, 1]$. Por otro lado, el centro ha desarrollado un indicador de forma física que recoge todos los aspectos relevantes en un único valor, $f_j, j = 1, \dots, n$, con $f_j \in [0, 1]$. Para incentivar la actividad, el precio que

1.5 Grafos parciales con aristas negativas

carga el centro es inversamente proporcional al nivel de forma física del cliente. En cada periodo, los n entrenadores son asignados a n clientes, formando así n pares (i, j) , donde i es el entrenador y j el cliente que se le ha asignado. El importe que factura el centro se calcula a partir de esta asignación: para cada par (i, j) se calcula el término $t_i \cdot (1 - f_j)$. La facturación total es la suma de estos términos para los n pares.

Se pide encontrar una asignación completa de entrenadores-clientes con una facturación asociada máxima.

1.5. Grafos parciales con aristas negativas

Consideramos como entrada un grafo G , conexo y ponderado, de modo que las aristas tienen como posibles pesos valores reales (\mathbb{R}). Se desea encontrar un grafo parcial conexo de G , (que incluya todos los vértices) tal que la suma de las aristas incluidas en el grafo parcial sea lo más pequeña posible.

1.6. El viajante de comercio (PVC)

Una empresa que comercializa componentes informáticos ha contratado un nuevo comercial. Al agente le ha asignado un conjunto de clientes, facilitándole la información relevante de cada uno, que incluye nombre, correo electrónico, teléfono y dirección postal. El director comercial le encarga al nuevo agente visitar a todos esos clientes, con el objetivo de hacer buenas ventas y que no vuelva hasta que haya visitado a todos los clientes. Por tanto, el recorrido considerado debe comenzar y finalizar en la empresa. Además, sólo podrá visitar cada cliente una sola vez. Además, para reducir gastos de desplazamiento, la distancia total recorrida debe ser lo más corta posible.

2. Trabajo a realizar

Para alcanzar el objetivo previsto con esta práctica, los alumnos deberán realizar las tareas que se detallan en la sección siguiente, documentando el trabajo realizado en una memoria con el contenido y estructura que se indica en la subsección 2.2.

2.1. Tareas

1. Estudiar en profundidad el problema asignado, asegurándose de comprender bien las entradas y salidas involucradas, así como las relaciones

entre éstas. Deberán identificar el tipo de información que define cada caso del problema y las magnitudes que definen su tamaño.

2. Diseñar un algoritmo voraz para resolver el problema.
3. Demostrar o justificar la validez del algoritmo, es decir, que la solución que proporciona es la correcta (óptima en el caso de problemas de optimización).
4. Implementar el algoritmo.
5. Diseñar e implementar un generador de casos para el problema asignado.
6. Realizar el análisis de la eficiencia teórica, empírica e híbrida los algoritmos.
7. Para el problema del “viajante de comercio”, habrá que diseñar tres métodos aproximados (heurísticas). El estudio de estos métodos incluirá su diseño, explicando por qué funcionan, su implementación y estudio de análisis de eficiencia.
8. Diseñar e implementar un generador de casos para el PVC.
9. Elaborar una memoria que documente el trabajo realizado de acuerdo a las indicaciones incluidas en la subsección 2.2.

2.2. Memoria

Todo el trabajo realizado debe redactarse en una memoria. La estructura de este documento será la siguiente:

1. Portada. Incluyendo las denominaciones de titulación, asignatura y práctica. También el nombre completo de los alumnos que forman el grupo y su dirección de correo electrónico.
2. Autores. Indicar el % del trabajo realizado por cada alumno, especificando qué tareas ha realizado cada uno.
3. Objetivos. Descripción del objetivo de la práctica.
4. Definición del problema. Descripción de los casos usados en la evaluación de la eficiencia. Descripción completa del entorno de análisis: hardware, sistema operativo, compilador, etc. empleados. Descripción del método de medición de tiempos.

5. Algoritmo diseñado. Descripción del algoritmo. Demostración o justificación de su validez.
6. Análisis de la eficiencia: teórica, empírica e híbrida.
Problema del Viajante de comercio
7. Diseño de la primera heurística para el PVC. Justificación de su validez.
8. Diseño de la segunda heurística para el PVC. Justificación de su validez.
9. Diseño de la tercera heurística para el PVC. Justificación de su validez.
10. Análisis comparativo de las tres heurísticas en términos de la calidad de las soluciones que construyen y tiempos de ejecución.
11. Conclusiones.

3. Evaluación de la práctica

Esta práctica se realizará por grupos formados por tres alumnos, ya conformados para las prácticas anteriores.

El profesor asignará a cada grupo el problema que deben resolver.

Los alumnos habrán de entregar dos archivos en la actividad correspondiente incluida en la página correspondiente a la asignatura en la plataforma Prado. El primero archivo será la memoria, en formato pdf.

El segundo será el código fuente de las implementaciones realizadas empaquetado en un fichero .zip. El código estará organizado en seis carpetas de nombres:

1. **Generador**: generador de casos del problema asignado.
2. **Voraz**: algoritmo que resuelve el problema asignado.
3. **Generador-pvc**: generador de casos para el problema del viajante de comercio.
4. **Pvc1**: Heurística 1 para resolver el PVC.
5. **Pvc2**: Heurística 2 para resolver el PVC.
6. **Pvc3**: Heurística 3 para resolver el PVC.

3. Evaluación de la práctica

Cada una de estas carpetas incluirá todos los módulos de código fuente necesarios para generar el programa binario correspondiente. Además, en la carpeta raíz habrá un único fichero **Makefile** cuyo objetivo por defecto construirá los seis binarios de nombres **generador**, **voraz**, **generador-pvc**, **pvc1**, **pvc2** y **pvc3**. Los programas 2, 4, 5, y 6 aceptarán un único argumento en línea de órdenes, que será el nombre de un fichero que incluirá los datos de un caso y enviarán a la salida estándar la solución calculada.

La fecha límite para entregar la memoria es el día **25 de abril de 2023** a las 23:59 horas.

Además de hacer el trabajo y entregar la memoria, cada equipo tendrá que elaborar una breve presentación del trabajo realizado que expondrá públicamente en clase de prácticas, de acuerdo a la convocatoria establecida por el profesor. Es obligatoria la participación de todos los miembros del grupo en la exposición.