UNIVERSIDAD DE GRANADA

Algorítmica

2º Grado en Ingeniería Informática

Guión de prácticas
Práctica 3

Algoritmos Voraces

© 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 Voraces

1. Objetivo

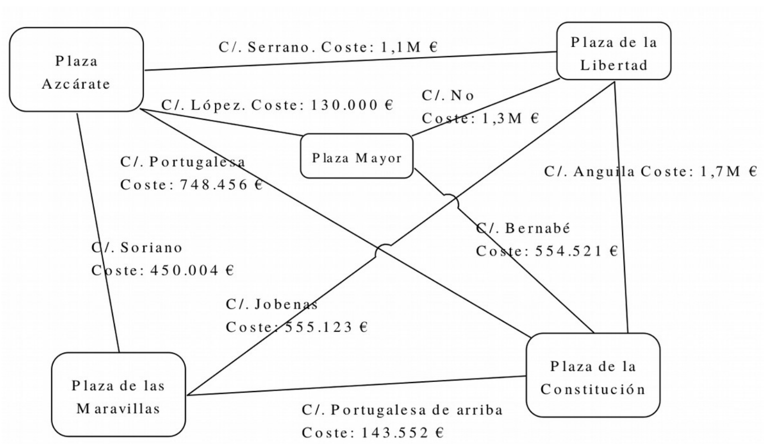
El objetivo de la práctica consiste en que el alumno sea capaz de analizar un problema y plantear una solución al mismo mediante la técnica de diseño de algoritmos voraces. Para ello, se propone un ejercicio guiado, a resolver en clase por el profesor, y un conjunto de ejercicios de los cuales el alumno deberá resolver **un ejercicio únicamente, con carácter individual o por parejas.**

2. Ejercicio guiado

Este ejercicio será resuelto en clase de prácticas por el profesor. El alumno deberá atender a la metodología de diseño seguida, para realizar posteriormente los mismos pasos en el diseño de un algoritmo que resuelva un problema del apartado 3 de este documento.

El alcalde de “Algovilla del Tuerto”, un conocido pueblo, desea renovar el embaldosado de las calles de su localidad. Sin embargo, las arcas del ayuntamiento no están muy saneadas y no se puede permitir embaldosar todas las calles del pueblo. El encanto de Algovilla reside en sus múltiples plazas (una en cada intersección de calles), que son monumentos locales, y en la belleza de pasear entre dichas plazas en época de verano, por lo que es un atractivo turístico que trae riqueza y trabajo a la localidad en esta época del año. El no realizar el embaldosado puede disminuir el atractivo del pueblo, la visita turística y, por tanto, reducir el desahogo económico que se produce en verano para sus habitantes, debido al turismo.

Para solucionar el problema, el concejal de urbanismo ha propuesto la siguiente solución: Asfaltar el mínimo número de calles, siempre que se pueda llegar desde una plaza a cualquier otra a través de calles asfaltadas. Así, cualquier turista podrá disfrutar de los paseos entre estos monumentos. Como asesor, se te requiere que formules el problema y lo resuelvas, proporcionando una solución que permita viajar desde cualquier plaza a cualquier otra plaza, con la restricción de que el paseo se realice siempre por una calle asfaltada y que el coste de asfaltar las calles necesarias para ello sea mínimo. Se te proporcionará información sobre qué plazas están unidas ntre sí directamente por una única calle y el coste de asfaltarlas (suponer coste igual a +∞ cuando no exista una calle que une dos plazas de forma directa). Un ejemplo serían las siguientes plazas, calles y costes:



La solución proporcionada por el profesor hará especial énfasis en los siguientes puntos:

- Analizar y describir si el problema es resoluble mediante la técnica de diseño voraz de algoritmos (es problema de optimización, por etapas, existe lista de candidatos, etc.).
- Diseñar cada componente greedy para solucionar el problema.
- Adaptar el esquema general greedy (plantilla del algoritmo) para solucionar el problema.
- Justificar el diseño realizado y explicar, mediante un ejemplo simple, cómo sería el funcionamiento del algoritmo diseñado.
- Demostrar si el algoritmo devuelve la solución óptima al problema o, en caso de no ser posible, exponer un contraejemplo que ponga de manifiesto la no optimalidad.
- Indicar el orden de eficiencia del algoritmo (en el caso peor), indicando las variables de las que depende el tamaño de caso del problema.
- Proporcionar una implementación del algoritmo, junto con un ejemplo de su uso.

3. Ejercicios propuestos

El alumno deberá resolver **únicamente uno** de los siguientes problemas. La asignación del problema se realizará en clase por parte del profesor. Siga la misma metodología explicada por el profesor de prácticas para resolver problemas Greedy.

Se permite que el alumno tenga iniciativa para realizar todas las suposiciones pertinentes que desee para el problema a resolver, siempre y cuando dichas suposiciones, y las decisiones derivadas de las mismas, no contradigan el enunciado del ejercicio. Los problemas tienen un carácter abierto, dando pie a múltiples formas de abordar cada uno de ellos. De esta forma, se fomentará la creatividad, la iniciativa y la actitud crítica requeridas en las competencias básicas de la asignatura.

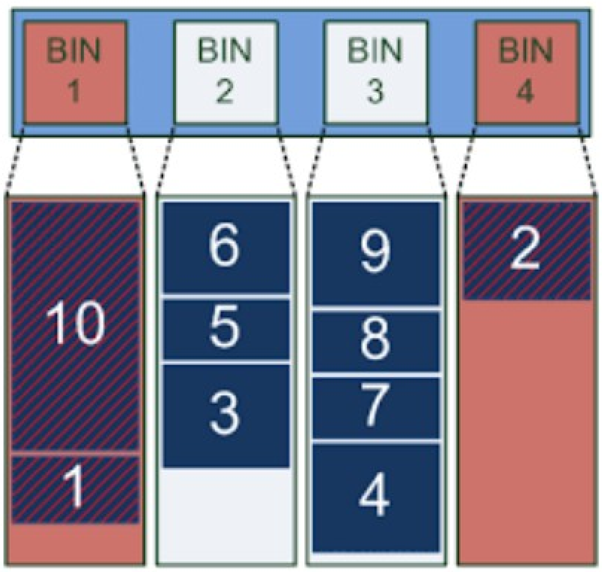
Ejercicio 1

Una agencia de viajes ha puesto en su web un sistema que calcula las rutas de autobús que debe tomar un pasajero para viajar entre dos ciudades seleccionadas. En algunos casos no existen rutas directas, por lo que es posible que el pasajero deba parar en algunas estaciones para hacer trasbordo. Para simplificar el problema, supondremos que siempre es posible realizar trasbordo. Además, asumimos que se conoce inicialmente el conjunto de ciudades conectadas directamente entre sí, así como el úmero de kilómetros que une cada par de ciudades para las que existe ruta directa entre ellas. La siguiente ilustración muestra un ejemplo de esta situación de partida:

El objetivo de la práctica consiste en elaborar un algoritmo greedy que permita conocer la ruta que debe seguir un viajero para ir desde una ciudad origen a otra destino con un número de kilómetros mínimo.

Ejercicio 2

Problema de Bin Packing. Se tiene un conjunto de N objetos indexados de 1 a N, cada uno con un volumen. Se dispone de tantos embalajes como queramos, todos del mismo volumen V. ¿Cómo insertamos los objetos para minimizar el número de embalajes necesarios?



Ejercicio 3

Se han organizado cursos de esquí en Sierra Nevada para los que se dispone de esquíes de diversos tamaños. A cada alumno se le deben asignar unos esquís cuyo tamaño sea lo más cercano posible a su altura. Sabiendo las alturas de los alumnos y los tamaños de los esquís disponibles, es necesario hacer una asignación de los mismos tal que el error cuadrático medio sea mínimo. Suponiendo que hay N alumnos y M esquís de longitud L(j) ($1 \leq j \leq M$), H(i) es la altura del alumno *i*, A(i) es la longitud del esquí asignado al alumno *i*, el error cuadrático medio se calcula de la siguiente forma:

$$E = \frac{1}{N}\sum_{i=1}^{N}(H(i) - L(i))^2$$

Ejercicio 4

Tenemos que completar un conjunto de n tareas con plazos límite. Cada una de las tareas consume la misma cantidad de tiempo (una unidad) y, en un instante determinado, podemos realizar únicamente una tarea. La tarea i tiene como plazo límite d_i y produce un beneficio g_i ($g_i > 0$) sólo si la tarea se realiza en un instante de tiempo $t \leq d_i$. Diseñe un algoritmo greedy que nos permita seleccionar el conjunto de tareas que nos asegure el mayor beneficio posible.

Ejercicio 5

Consideremos un grafo no dirigido G = (V, E). Un conjunto de nodos U se dice que es un recubrimiento de G si U es un subconjunto de V tal que cada arista en E incide en, al menos, un vértice de U. Un conjunto de nodos es un recubrimiento minimal de G si es un recubrimiento con el número mínimo posible de nodos. Diseñe un algoritmo greedy para obtener un recubrimiento de G.

Ejercicio 6

Supongamos que disponemos de n trabajadores y n tareas. Sea $c_{ij} > 0$ el coste de asignarle la tarea j al trabajador i. Una asignación válida es aquella en la que a cada trabajador le corresponde una tarea y cada tarea la realiza un trabajador diferente. Dada una asignación válida, definimos el coste de dicha asignación como la suma total de los costes individuales. Diseñe un algoritmo greedy para obtener una asignación de tareas a trabajadores óptima.

Ejercicio 7

Se desea diseñar un nuevo teclado de PC para un nuevo idioma. Sabiendo que f_{ab} es la frecuencia en el que una letra a del idioma precede a otra letra b, y d_{ij} es la distancia física que existe entre dos teclas i y j del teclado, diseñe un algoritmo Greedy que asigne una letra del alfabeto a cada tecla de modo que se minimice la siguiente expresión:

$$p=\min_p\{H(p)\}=\min_p\{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f_{p(i)p(j)} d_{ij}\}$$

donde p(i) es la letra del alfabeto asignada a la tecla i del teclado.

Ejercicio 8

Dada una secuencia de palabras p_1 , p_2 , ..., p_n de longitudes l_1 , l_2 , ..., l_n se desea agruparlas en líneas de longitud L. Las palabras están separadas por espacios cuya amplitud ideal (en milímetros) es b, pero los espacios pueden reducirse o ampliarse si es necesario (aunque sin solapamiento de palabras), de tal forma que una línea conteniendo las palabras p_i , p_{i+1} , ..., p_j tenga exactamente longitud L. Sin embargo, existe una penalización por reducción o ampliación en el número total de espacios que aparecen o desaparecen. El costo de fijar la línea p_i , p_{i+1} , ..., p_j es (j – i)|b' – b|, siendo b' el ancho real de los espacios, es decir (L – l_i – l_{i+1} – ... – l_j)/(j – i). No obstante, si j = n (la última palabra) el costo será cero a menos que b' < b (ya que no es necesario ampliar la última línea). Diseñar un algoritmo greedy que divida un conjunto de palabras en líneas minimizando el coste total de arreglar los espacios en cada línea.

Ejercicio 9

Un camionero conduce desde una ciudad A a otra ciudad B siguiendo una ruta dada y llevando un camión que le permite, con el tanque de gasolina lleno, recorrer n kilómetros sin parar. El camionero dispone de un mapa de carreteras que le indica las distancias entre las gasolineras que hay en su ruta. Como va con prisa, el camionero desea pararse a repostar el menor número de veces posible. Deseamos diseñar un algoritmo greedy para determinar en qué gasolineras tiene que parar, minimizando el número de paradas.

Ejercicio 10

Sea G un grafo no dirigido conexo. Entonces G es un grafo de Euler si, y sólo si, todo los vértices de G tienen grado par. Recordemos que el grado de un vértice es el número de aristas que empiezan (o terminan, según se mire) en él. Un camino de Euler es un camino que no repite aristas y en el que aparecen todos las aristas del grafo. Un circuito de Euler es un ciclo (empieza y termina en el mismo vértice) que, además, es un camino de Euler. Sabiendo esto, diseñe un algoritmo Greedy que, teniendo como entrada un grafo no dirigido de Euler, proporcione como salida un circuito de Euler que recorra todas las aristas del grafo sin repetir (algoritmo de Fleury).

4. Entrega de la práctica

Se deberá entregar un documento **individual o por parejas** (memoria de prácticas) que contenga los siguientes apartados:

1. Análisis: Realizar una búsqueda bibliográfica del problema a resolver, su formulación teórica y posibles alternativas de solución. Indicar si el problema es resoluble mediante la técnica de algoritmos voraces. Describir el problema, qué decisiones se han tomado para abordarlo o acotarlo, y justifíquelas.
2. Diseño (I): Especificar las componentes Greedy del algoritmo, justificando su selección.
3. Diseño (II): Explique cómo adaptar la plantilla de la técnica Greedy vista en teoría para la resolución del problema, utilizando las componentes diseñadas en el apartado anterior. Exponga el algoritmo Greedy final.
4. Implementación: Indique cómo se han implementado cada una de las componentes del algoritmo diseñado en el apartado anterior. Adjunte a la memoria de prácticas, en un fichero separado, el código fuente. Indique en este apartado también cómo compilar el código y cómo ejecutarlo.
5. Pruebas: Proporcione un ejemplo de ejecución del algoritmo. Indique cómo ejecutar el código proporcionado para resolver los ejemplos. Explique, a mano, cómo funciona el algoritmo resolviendo una traza del algoritmo para el caso de ejemplo dado.

Estos apartados deberán contener las soluciones en el formato explicado en clase por el profesor.

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 se valorará de 0 a 10. Cada apartado se valorará con 2 puntos. Se valorará no sólo la exactitud y bondad de los resultados obtenidos, sino principalmente la explicación que realice el alumno sobre cómo ha realizado los pasos requeridos en cada apartado.

La práctica contribuirá con 3 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, realizará defensas 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.