EDA - II

Práctica 4

Backtracking

1. Objetivos.

- ✓ Construir la solución de un problema concreto derivado de las prácticas anteriores utilizando el esquema de la programación con "backtracking". El problema de aplicación utiliza como base el ejemplo de las prácticas anteriores.
- ✓ Realizar el análisis de la eficiencia de la solución aportada, tanto desde el punto de vista teórico como práctico.

2. Definición del problema.

En las distintas estaciones de metro de la red de la "Metropolitan Transport Authority" (NAMTA) de la expansión urbana de Nueva Almería hay máquinas expendedoras de tarjetas de un solo viaje y de recarga de las tarjetas de usuario. Dichas máquinas pueden estar en los siguientes estados:

a/ Activas (A), lo que significa que disponen de tarjetas unitarias para suministrar y de cambio para devolver cuando se compra o recarga con una cantidad una tarjeta.
b/ Bloqueadas (B), cuando no disponen de tarjetas para emitir o no disponen de cambio para hacer las operaciones normales.

En la realidad hay más estados, ya que pueden trabajar con tarjetas de crédito (por lo que no importa la situación del cambio), o pueden sólo recargar tarjetas (con tarjeta o con efectivo). Con objeto de simplificar la práctica vamos a considerar sólo los estados A y B.

Las estaciones de metro disponen de una cantidad de máquinas expendedoras dada por la categoría de la estación según una tabla de valores de entrada, como la que sigue:

Tipo de estación	***Número de máquinas***	***Ejemplo***
Especiales: Em & Ce	ne	30
Cruces/intercambiadores	ni	20
Central and Circle (líneas 1 y 2)	nc	12
Normal (líneas 3-6)	nn	10

Al inicio de la jornada, cada estación de metro E(a,c) envía la información de la situación de sus máquinas (mediante un mensaje al sistema de control central), en el que indica:

- ✓ nA – Número de máquinas activas
- ✓ nB – Número de máquinas bloqueadas

Una estación E(a,c) puede estar en dos estados:

- ✓ No automatizada (NA), cuando el número de máquinas bloqueadas es igual al número de máquinas que tiene. No hay ninguna máquina completamente operativa. Los clientes tienen que ir a ventanilla para operar, lo que significa grandes colas.
- ✓ Automatizada (SA), cuándo el número de máquinas bloqueadas es menor que el número de máquinas disponibles en la estación.

Por la mañana temprano, al abrir la red, se recibe en la oficina central la información de todas las estaciones sobre la situación de las máquinas. En ese momento se prepara un equipo de revisión/actualización de las máquinas (que carga tarjetas y/o cambio, y repara las máquinas que lo requieran). A efectos de simplicidad vamos a considerar un único equipo (en la realidad podría haber más de uno).

El equipo va a utilizar la red de metro para moverse entre las distintas estaciones, con lo que el tiempo de desplazamiento entre estaciones es el que corresponde según se indica en la práctica 2 parte b. La ubicación original del equipo de control de las máquinas está en la estación Central (Ce).

El tiempo medio en revisar/actualizar una máquina es de tm (ej. 2 minutos). El equipo revisa sólo las máquinas bloqueadas de la estaciones.

Las condiciones para hacer el recorrido son las siguientes:

a/ Se ha de minimizar el tiempo total de no automatización de estaciones. Dicho tiempo se calcula de la siguiente forma:

- Una estación está en estado NA desde el inicio de la jornada (ej. 4 a.m.) hasta que se actualizan todas sus máquinas. Ejemplo, con dos estaciones (E1, E2) en estado NA:
  - ✓ Supongamos que desde el inicio de la jornada el equipo tarda en llegar 25 minutos a la primera estación (E1), y tarda en revisar todas sus máquinas (ej. 10 máquinas), 20 minutos; el tiempo de no automatización (NA) de esa primera estación es de: 45 minutos.
  - ✓ A continuación va a la segunda estación (E2), tardando 30 minutos desde E1, dicha estación tiene por ejemplo 20 máquinas, tardando en revisarlas 40 minutos. Por tanto el tiempo de actuación en E2 es 30+40=70 minutos. Pero el tiempo total de espera es 45 minutos (que el equipo ha estado en la primera estación) más los 70 minutos de atender la segunda estación, total 115 minutos.

El tiempo total de espera (No automatización de estaciones) es de TE1+TE2 = 45+115= 160 minutos. Éste es el valor que se intenta minimizar.

b/ Una vez que no hay estaciones en estado NA, el equipo pasa a revisar las máquinas en estado B (bloqueadas) de las estaciones en estado SA. La idea es optimizar también el tiempo en que hay máquinas no activas. Se considera el tiempo de espera de CADA máquina en estado de bloqueada (B), haciendo los cálculos de forma similar al caso de las estaciones, pero máquina a máquina.

Primero se actúa sobre las estaciones en estado NA, minimizando el tiempo que hay estaciones No Activas, y luego se revisan las máquinas restantes bloqueadas minimizando el tiempo que están bloqueadas en conjunto (antes de actuar sobre las máquinas de estaciones activas – SA -, ha pasado TODO el tiempo de trabajar sobre las

estaciones NA, es un tiempo de espera de partida para todas las máquinas). Este segundo criterio de optimización se aplica máquina a máquina. Ha de tenerse en cuenta el tiempo de traslado entre estaciones, el tiempo de ir de una máquina a otra dentro de una estación se considera despreciable.

Se trata un problema de optimización con dos criterios, el primero – prioritario – es reducir el tiempo de No Actividad de las estaciones. Una vez resuelto ese problema, se reduce el tiempo acumulado de bloqueo de las máquinas.

Han de utilizar la información de las prácticas 2 y 3; puede que tengan que hacer una rectificación de alguna de ellas para disponer de los datos necesarios (por ejemplo en forma de tabla).

Se desea un plan de visitas para el equipo de revisión de las máquinas de gestión de tickets en las estaciones. No vamos a considerar restricciones horarias, y vamos a trabajar con un solo equipo, en la realidad sería más de uno para que diera tiempo a revisar "por la mañana" todas las máquinas. Pueden utilizar los valores de tiempo de ejemplo, convenientemente parametrizados (pueden ser almacenados como constantes con nombre).

3. Trabajo a desarrollar.

Deberán proponer e implementar la solución al problema de programar las visitas a todas las estaciones de la red, con vistas a evitar la existencia de estaciones en estado NA y, en segundo lugar, de máquinas en estado B. Minimizando los tiempos pertinentes (TNA, total de estaciones no activas y TMB, total de máquinas bloqueadas).

No olviden analizar si es necesario o conveniente algún tipo de **pre-procesamiento de los datos** que resultan de las prácticas anteriores (para optimizaar el formato de entrada o datos de soporte, por ejemplo) o que se han de generar para esta práctica, antes de entrar en el método backtracking de cada caso.

Analicen con detalle el **tipo de árbol para representar el espacio de exploración** de cada uno de los grupos de soluciones, **siendo en este problema de especial importancia pensar y justificar el formato de la solución** y las funciones objetivo o de optmización, así como las **funciones de poda** – que son de especial importancia para implementar de forma eficiente la solución backtracking.

Seguidamente, deberán de implantar dichos esquemas en Java (u otro lenguaje), probando su funcionamiento y comprobando experimentalmente el comportamiento del tiempo de ejecución.

Pueden realizar una implementación que utilice los resultados de las prácticas anteriores, aparte de leer los datos de entrada necesarios. Pueden trabajar con una única red de metro, valores (m,n), para evaluar la eficiencia. En este caso, cambia el número de estaciones en estado NA y de máquinas en estado B (dejamos fijo el tamaño de la red, para evitar complicaciones en el cálculo de la eficiencia).

4. Entregas.

Se ha de entregar, en fecha, un archivo zip o rar con toda la documentación requerida, lo que supone:

a) Una memoria explicativa del diseño del esquema algorítmico de "backtracking" que se plantea. Deben señalarse los elementos propios del esquema (por ejemplo, el árbol de estados y las **condiciones de poda**.) No olviden indicar cómo se utilizan las funciones de optimización.

b) Es importante señalar a cuál o cuáles de los ejemplos vistos en clase se parece/n al problema; indicando la familia a la que pertenecen y sus peculiaridades (cuidado, alguno de los problemas puede tener características "extras" sobre un problema estándar visto).

c) El modelo de clases correpondiente al esquema algorítmico diseñado, así como la selección de las estructuras de datos que soporten el procesamiento.

d) El código correspondiente documentado de la aplicación, desarrollada en Java (u otro lenguaje), que resuelva el problema propuesto. La aplicación tendrá las opciones oportunas para evaluar el coste temporal del proceso de forma experimental.

e) Los juegos de prueba (un mínimo de 10 juegos); tendrán que decidir los juegos que consideren representativos para evaluar correctamente la eficiencia de los algoritmos; también tendrán que entregar las salidas generadas.

f) Un análisis de la eficiencia del resultado.

El archivo entregado deberá estar libre de virus, la presencia de cualquier tipo de malware supondrá el suspenso automático de la asignatura. Si no disponen de herramientas de detección de calidad, pueden utilizar el servicio que se ofrece en las aulas de informática de control de virus.

