

# ANÁLISIS Y DISEÑO DE ALGORITMOS

---

## RAMIFICACIÓN Y PODA

### Práctica final

Entrega: Hasta el domingo 5 de junio, 23:55h. A través de Moodle

---

## Encaminamiento óptimo y IV

[ El enunciado del problema es idéntico a la práctica 8, lo que cambia es la estrategia de resolución, y algunos datos a mostrar. Por lo tanto, si ya conoces el enunciado puedes continuar leyendo en la página 2.]

Una empresa que ofrece servicios de almacenamiento en la nube dispone de  $n$  servidores (que también llamaremos nodos) donde aloja los archivos de sus clientes. **Los nodos están conectados entre sí mediante un sistema de comunicaciones bidireccional que conforman un grafo conexo no necesariamente completo.**<sup>1</sup>

Por cuestión de costes, solo una cantidad  $m$  de servidores ( $0 < m \leq n$ ) pueden estar conectados directamente con el exterior. Llamaremos “puerta de enlace” a un nodo de almacenamiento que dispone de conectividad exterior directa. Si un nodo no es puerta de enlace entonces se le asignará la que está más cerca (que será la puerta de enlace asociada a ese nodo sin conexión exterior directa). Cada archivo que un cliente solicita para su descarga se envía desde el nodo que lo aloja hasta su puerta de enlace asociada, teniendo que atravesar (si es el caso) todos los nodos intermedios que los separa.

Por otra parte, cada nodo  $i$  está caracterizado por un parámetro  $c_i \in \mathbb{R}^+$  que indica su capacidad de almacenamiento. Cuanto más alto es su valor, más archivos puede albergar el nodo.

Definimos *tráfico estimado* de una instalación como  $\sum_{i=0}^{n-1} c_i \cdot d_{ij}$ , donde  $d_{ij} \in \mathbb{R}^+$  es la distancia más corta que separa el nodo  $i$  de su puerta de enlace asociada (el nodo  $j$ ). Si el nodo  $i$  es, él mismo, puerta de enlace (es decir,  $i = j$ ), entonces  $d_{ij} = 0$  (por lo tanto, los nodos que son puerta de enlace no aumentan el tráfico estimado ya que no hacen uso del bus privado).

Se pretende seleccionar los  $m$  nodos que actuarán de puerta de enlace hacia el exterior y, para el resto de los nodos, indicar cuáles serán sus puertas de enlace asociadas. Todo ello con el objetivo de minimizar el tráfico estimado de la instalación (para simplificar, supondremos que el valor de  $m$  ha sido prefijado).

En cuanto a la forma de suministrar las distancias, para simplificar, el programa recibirá una matriz cuadrada  $d$ ,  $n \times n$ , donde  $d_{ii} = 0$  y  $d_{ij} = d_{ji} \in \mathbb{R}^+$  contiene la distancia del camino más corto entre el nodo  $i$  y el nodo  $j$ ,  $\forall i, j : 0 \leq i, j \leq n - 1$  (de esta manera no es relevante conocer cuál es el canal que produce el camino más corto o cuántos nodos intermedios se atraviesan).

Por otra parte, la capacidad de almacenamiento de cada nodo vendrá dada en un vector  $c$  con  $c_i \in \mathbb{R}$ .

---

<sup>1</sup>En el grafo, los vértices son los nodos y las aristas los canales de comunicación entre nodos; aunque no todos los pares de vértices del grafo estén conectados por aristas, se garantiza que todos los vértices son alcanzables entre sí (grafo conexo).

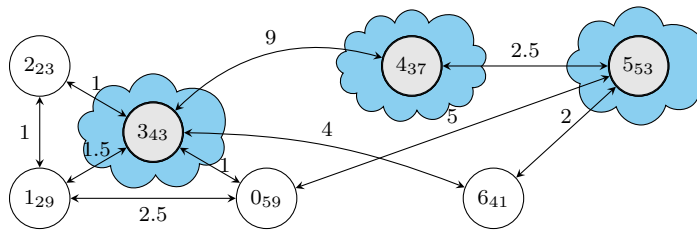


Figura 1: Posible ubicación de las tres puertas de enlace en los nodos dibujados sobre una nube. Los archivos de los nodos 0, 1 y 2 irían al nodo 3 (ya que es la puerta de enlace más cercana), y los del nodo 6 irían a la puerta de enlace ubicada en el nodo 5. El tráfico estimado asociado a esta solución viene dado por la expresión  $59 \cdot 1 + 29 \cdot 1,5 + 23 \cdot 1 + 41 \cdot 2 = 207,5$ . **Esta ubicación de las puertas de enlace es la mejor.**

En esta práctica se pide aplicar el método ramificación y poda (*Branch & bound*) para obtener la mejor ubicación posible de las puertas de enlace.

■ **Nombre del programa y sintaxis de la orden.**

El programa a realizar se debe llamar **met\_bb**. La orden tendrá la siguiente sintaxis:

```
met_bb -f fichero_entrada
```

El problema a resolver se suministrará codificado en un fichero de texto cuyo nombre se recogerá a través de la opción **-f**. Su formato y contenido será:

- Línea 1 del fichero: Valores  $n$  y  $m$ , en este orden.
- Línea 2: Capacidad de almacenamiento de cada nodo: Una lista de  $n$  números reales positivos incluyendo el 0.
- Línea 3 y  $n - 1$  líneas siguientes: Una matriz simétrica de números reales positivos que contiene el camino más corto entre cualquier par de nodos según se ha descrito anteriormente.

Por tanto, el fichero contendrá  $n + 2$  líneas que finalizarán con un salto de línea, salvo, en todo caso, la última línea. El carácter separador entre números siempre será el espacio en blanco.

El programa debe ser robusto en cuanto al uso incorrecto de la sintaxis de la orden. En este sentido, debe informar también del primer error que se detecte (siguiendo las directrices de las prácticas anteriores). No es necesario controlar posibles errores en el fichero de entrada ya que este siempre se ajustará fielmente al formato establecido.

A través de *Moodle* se puede descargar un archivo comprimido con varios ejemplos y sus soluciones; entre ellos está **07n03m.p**, utilizado como ejemplo en este enunciado.

■ **Salida del programa, formato de salida y ejemplo de ejecución.**

El programa mostrará los siguientes cuatro resultados, cada uno en una línea distinta:

1. Línea 1: Tráfico estimado que se obtiene de la ubicación escogida para las puertas de enlace.
2. Línea 2: La relación de los  $m$  nodos que han sido seleccionados para actuar de puerta de enlace, separados por un espacio en blanco.<sup>2</sup>

<sup>2</sup>No importa el orden en el que se muestren los nodos seleccionados. Por otra parte, la selección que produce un tráfico estimado mínimo puede no ser única; en estos casos, da igual cuál se muestre.

3. Línea 3: Una cuantificación de los distintos tipos de nodos encontrados durante el proceso de búsqueda de la solución óptima. Se mostrará los siguientes valores, **en este orden** y separados mediante un espacio en blanco:
  - a) Número de nodos visitados,
  - b) Número de nodos explorados,
  - c) Número de nodos completados (nodos hoja),
  - d) Número de nodos descartados por no ser factibles,
  - e) Número de nodos descartados por no ser prometedores,
  - f) Número de nodos que fueron prometedores pero que finalmente se descartaron,
  - g) Número de veces que la mejor solución hasta el momento se ha actualizado a partir de un nodo completado,
  - h) Número de veces que la mejor solución hasta el momento se ha actualizado a partir de la cota pesimista de un nodo sin completar.
4. Línea 4: El tiempo de CPU, expresado en milisegundos, que se ha consumido para resolver el problema. Si se utiliza un subóptimo de partida o se hace una preparación previa de los datos, el tiempo necesario para ello también cuenta en este cómputo.

Cada uno de los cuatro tipos de resultados enumerados se mostrará en una línea distinta, que terminará con un único salto de línea y nada más. Por lo tanto, los valores que corresponden a la estadística sobre los nodos, que se refieren al árbol de búsqueda de ramificación y poda, se mostrarán en una única línea y separados mediante un espacio en blanco. Debe tenerse en cuenta que estos estadísticos dependen de cada implementación particular. Puede darse el caso de que no proceda mostrar el valor de un elemento concreto (cuando, por ejemplo, no se hace uso de algunas de las cotas, etc.), en tal caso se sustituirá el valor por un guion medio ('-').

Por ejemplo:

```
$met_bb -f 07n03m.p
207.5
4 3 5
28 14 0 0 14 0 0 2
0.03
```

Es imprescindible ceñirse al formato de salida mostrado. La última línea debe terminar con un salto de línea (y sólo uno). **En ningún caso debe añadirse texto o valores adicionales.**

#### ■ Documentación del trabajo realizado.

Se debe entregar una **memoria**, en formato PDF, con nombre **memoria.pdf**, que describa las estructuras de datos empleadas así como la/s estrategia/s de búsqueda, los mecanismos de poda y las cotas optimistas y pesimistas utilizadas.

Incluirá también un análisis del comportamiento de distintas estrategias de búsqueda combinándolas, si se considera apropiado, con distintos mecanismos de poda. Para ello, se realizará un cuadro comparativo que contenga, para cada caso estudiado, el número de nodos que se añaden a la lista de nodos vivos y el tiempo de respuesta del algoritmo (en las diapositivas de teoría se muestran cuadros comparativos que pueden servirte de guía).

La primera línea de la memoria contendrá el nombre del autor y su DNI. El resto se estructurará en seis apartados de la siguiente manera:

## 1. Estructuras de datos

### 1.1 Nodo

En este apartado se describirá brevemente el contenido del nodo, es decir, las variables que lo componen y su cometido.

### 1.2 Lista de nodos vivos

Breve descripción de la estructura de datos utilizada y si procede, qué criterios se han analizado para extraer el nodo más prometedor (El programa debe utilizar el que se considere más rápido pero todos los demás se mencionarán también en este apartado).

## 2. Mecanismos de poda

### 2.1. Poda de nodos no factibles

Se deberá indicar cómo se descartan los nodos que no son factibles; si no procede este tipo de poda debe indicarse el motivo.

### 2.2. Poda de nodos no prometedores

Se deberá indicar cómo se decide que un nodo no es prometedor; si no procede debe indicarse el motivo.

## 3. Cotas pesimistas y optimistas

Se describirá en qué consisten dichas cotas, distinguiéndose, en el caso de la pesimista, entre el nodo inicial y los demás nodos.

### 3.1 Cota pesimista inicial (inicialización).

### 3.2 Cota pesimista del resto de nodos.

### 3.3 Cota optimista.

## 4. Otros medios empleados para acelerar la búsqueda

En este apartado se citará cualquier otro mecanismo utilizado con el objetivo de acelerar la búsqueda o cualquier otro aspecto que se quiera reflejar en la memoria y que no se adapta a ninguno de los apartados anteriores.

## 5. Estudio comparativo de distintas estrategias de búsqueda

En este apartado se incluirá el cuadro comparativo mencionado anteriormente junto con una breve discusión de los resultados obtenidos.

## 6. Tiempos de ejecución.

Se mostrará en este apartado el tiempo de proceso requerido para resolver los problemas de test que se relacionan, escogidos de entre los publicados. Si alguno de ellos no ha podido ser resuelto por tardar demasiado (máximo 5 minutos), se escribirá '?'. Por ejemplo:

- Fichero 30n12m.p: 1572.34 ms.
- Fichero 30n26m.p: 0.74 ms.
- Fichero 40n05m.p: 1230.53 ms.
- Fichero 40n10m.p: ?
- Fichero 40n20m.p: 110474.22 ms.
- Fichero 40n25m.p: 4123.63 ms.
- Fichero 50n05m.p: 7748.08 ms.
- Fichero 50n07m.p: ?

**IMPORTANTE:** Todos los apartados (salvo el último) se complementarán con el correspondiente fragmento de código que corrobore el trabajo descrito. Si algunos de los apartados no ha sido realizado se pondrá únicamente la frase “NO IMPLEMENTADO”.

De los distintos elementos que puede contener un algoritmo de ramificación y poda, sólo se evaluarán los que estén reflejados en esta memoria. Por lo tanto, todo lo que no esté en dicho documento se entenderá que no ha sido realizado.

■ **Sobre la evaluación de esta práctica.**

En la evaluación de este trabajo se tendrá en cuenta la calidad y organización del código fuente, las estructuras de datos utilizadas, el análisis de distintas estrategias y los mecanismos de poda implementados para minorar la cantidad de nodos explorados.

**No obstante, para que esta práctica sea evaluada es imprescindible que funcione con algunos ejemplos básicos, donde el tiempo de ejecución es irrelevante.**

Otro aspecto muy relevante en la nota final será el tiempo de respuesta, por lo tanto, aunque se hayan implementado distintas estrategias de búsqueda, funciones de cota, etc, **el programa que se entrega debe funcionar con la configuración que se entienda más rápida** (en la memoria se deberá hacer mención a cualquier otro trabajo realizado que, por ese motivo, no se haya incluido en la versión final del programa, aunque si este es el caso debe aparecer comentado en el código). En este sentido, es recomendable que incluyas, en el makefile, la opción del compilador `-O3` (optimizador de código).

■ **Normas para la entrega.**

**ATENCIÓN:** Estas normas son de obligado cumplimiento para que esta práctica sea evaluada.

Es imprescindible ceñirse al formato y texto de salida descrito, incluso en lo que se refiere a los saltos de línea o carácter separador, que en todos los casos es el espacio en blanco. No debe añadirse texto o valores adicionales. Si estos requisitos no se cumplen, es posible que falle una de las fases de la evaluación de este trabajo, que se hace de manera automática y en tal caso la práctica tampoco será evaluada.

1. El programa debe realizarse en un único archivo fuente con nombre `met_bb.cc`.
2. Se debe entregar únicamente los ficheros `memoria.pdf`; `met_bb.cc` y `makefile` (para generar el archivo ejecutable a través de la orden `make` sin añadir nada más). **Sigue escrupulosamente los nombres de ficheros que se citan en este enunciado. Solo hay que entregar esos tres archivos (en ningún caso se entregarán archivos de *test*).**
3. Es imprescindible que no presente errores ni de compilación ni de interpretación (según corresponda), en los ordenadores del laboratorio asignado.<sup>3</sup> Se tratará de evitar también cualquier tipo de *warning*.
4. Todos los ficheros que se entregan deben contener el nombre del autor y su DNI (o NIE) en su primera línea (entre comentarios apropiados según el tipo de archivo).
5. Se comprimirán en un archivo `.tar.gz` cuyo nombre será el DNI del alumno, compuesto de 8 dígitos y una letra (o NIE, compuesto de una letra seguida de 7 dígitos y otra letra). Por ejemplo: `12345678A.tar.gz` o `X1234567A.tar.gz`. **Solo se admite este formato de compresión y solo es válida esta forma de nombrar el archivo.**
6. En el archivo comprimido **no debe haber subcarpetas**, es decir, al extraer sus archivos, estos deben quedar guardados en la misma carpeta donde está el archivo que los contiene.
7. La práctica hay que subirla a *Moodle* respetando la fecha expuesta en el encabezado de este enunciado.

---

<sup>3</sup>Si trabajas con tu propio ordenador o con otro sistema operativo asegúrate de que este requisito se cumple (puedes comprobarlo haciendo uso del compilador *online* de <https://godbolt.org>).