

ANÁLISIS Y DISEÑO DE ALGORITMOS

RAMIFICACIÓN Y PODA

Práctica final

Entrega: Hasta el domingo 30 de mayo, 23:55h. A través de Moodle

La planificación del alfarero (y IV)

[El enunciado del problema es idéntico a la práctica 8, lo que cambia es la estrategia de resolución, los datos a mostrar y el formato de salida]

Un alfarero desea planificar la temporada de producción teniendo en cuenta un repertorio de n objetos que sabe elaborar y un tiempo máximo $T \in \mathbb{R}$ que puede dedicar a su fabricación. De cada objeto i se conoce el tiempo t_i que tarda en realizarlo y el valor v_i al que lo vende, donde $t_i \in \mathbb{R}$ y $v_i \in \mathbb{N}$, $\forall i : 1 \leq i \leq n$. Por otra parte, el número de copias que se puede fabricar de cada objeto i está limitado por $m_i \in \mathbb{N}$ y solo puede vender objetos que están terminados.

Por ejemplo, para los valores $n = 4$; $T = 11.02$; $t = (4.7, 4.4, 1.2, 8.9)$; $v = (5, 10, 4, 6)$ y $m = (8, 8, 7, 4)$, la solución óptima a este caso particular corresponde con fabricar una copia del segundo objeto; cinco del tercero y ninguna de los restantes, cuya fabricación consumiría 10.4 unidades de tiempo y produciría una ganancia de valor 30.

En esta práctica se pide aplicar el método ramificación y poda (*Branch & bound*) para encontrar la máxima ganancia que puede obtener el alfarero.

1. Nombre del programa, opciones y sintaxis de la orden.

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

potter_bb -f fichero_entrada

- La instancia del problema a resolver se suministrará a través de un fichero de texto cuyo nombre se recogerá a través del parámetro obligatorio -f.

El formato y contenido del mencionado fichero es idéntico a la práctica anterior:

- Línea 1 del fichero: Valores n (número entero positivo no nulo) y T (número real positivo no nulo) separados mediante un único espacio en blanco.
- Línea 2, 3 y 4: los vectores t , v y m , uno por cada línea y en ese orden. El separador es el espacio en blanco (sólo uno). La línea que corresponde al vector t contendrá n números reales positivos no nulos mientras que las restantes dos líneas (vectores v y m) contendrán n números enteros positivos

Por lo tanto, el fichero contendrá 4 líneas que finalizarán con un salto de línea, salvo en todo caso, la última línea.

A través de *Moodle* se puede descargar un archivo comprimido con varios ejemplos junto con las soluciones, entre ellos está **potter_n04r.def** utilizado como ejemplo en este enunciado.

2. Salida del programa, formato de salida y ejemplo de ejecución.

3. Salida del programa y formato de salida

Los resultados que el programa debe mostrar se organizarán en tres líneas. No se debe incorporar texto alguno salvo el propio valor numérico (véase los ejemplos de ejecución para obtener información acerca de la sintaxis de salida):

- a) En la primera línea, la ganancia del alfarero con la mejor planificación posible.
- b) En la segunda, 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:
 - 1) Número de nodos visitados,
 - 2) Número de nodos explorados,
 - 3) Número de nodos hoja visitados,
 - 4) Número de nodos descartados por no ser factibles,
 - 5) Número de nodos descartados por no ser prometedores,
 - 6) Número de nodos que fueron prometedores pero que finalmente se descartaron,
 - 7) Número de veces que la mejor solución hasta el momento se ha actualizado a partir de un nodo completado,
 - 8) Número de veces que la mejor solución hasta el momento se ha actualizado a partir de la cota pesimista de un nodo sin completar.
- c) En tercer y último lugar, el tiempo de CPU, expresado en milisegundos, que ha consumido el algoritmo de ramificación y poda para resolver el problema.

Cada uno de los tres 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:

```
potter_bb -f potter_n04r.def
30
64 10 1 47 7 1 1 1
0.003
```

4. 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 (según se ha explicado en clase de teoría).

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 cada uno de los ficheros de test publicados para esta práctica. Si alguno de los ficheros no ha podido ser correctamente resuelto por el programa realizado se escribirá '?'. Por ejemplo:

- Fichero potter_n50.def: 10.125 ms.
- Fichero potter_n60.def: 0.474 ms.
- Fichero potter_n70.def: 1000.38 ms.
- Fichero potter_n80.def: 1882.95 ms.
- Fichero potter_n90.def: ?
- Fichero potter_n100.def: ?
- Fichero potter_n120.def: ?

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.

5. 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).

Para valorar el tiempo de respuesta se hará una clasificación de todos los trabajos presentados. La nota en este apartado será proporcional a la situación en esta ordenación.

6. Normas para la entrega.

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

Es imprescindible ceñirse al formato de salida descrito (véanse también los ejemplos publicados a través de *Moodle*), incluso en lo que se refiere a los saltos de línea o carácter separador, que en todos los casos es el espacio en blanco. La última línea debe terminar con un salto de línea (y sólo uno). **En ningún caso debe añadirse texto o valores adicionales.** Deberá tratarse también posibles errores en los argumentos de la orden, no suministrar el fichero de entrada o su inexistencia, tal y como se hizo en la práctica anterior.

- a) El programa debe realizarse en un único archivo fuente con nombre `potter_bb.cc`.
- b) Se debe entregar únicamente los ficheros `potter_bb.cc`, `makefile` (para generar el archivo ejecutable) y `memoria.pdf`. Sigue escrupulosamente los nombres de ficheros que se citan en este enunciado. **No hay que entregar nada más (en ningún caso se entregarán archivos de test).**
- c) Es imprescindible que no presente errores ni de compilación ni de interpretación (según corresponda), en los ordenadores del laboratorio asignado y en el sistema operativo *Linux*.¹ Se tratará de evitar también cualquier tipo de *warning*.
- d) 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).
- e) 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.**
- f) En el archivo comprimido **no debe existir subcarpetas**, es decir, al extraer sus archivos estos deben quedar guardados en la misma carpeta donde está el archivo que los contiene.
- g) La práctica hay que subirla a *Moodle* respetando las fechas expuestas en el encabezado de este enunciado.

¹Si 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>).