

# ANÁLISIS Y DISEÑO DE ALGORITMOS

---

## VUELTA ATRÁS

### Práctica 8 de laboratorio

Entrega: Hasta el domingo 9 de mayo, 23:55h. A través de Moodle

---

## La planificación del alfarero III

[ Se trata del mismo problema que la práctica anterior con la diferencia de que ahora hay que obtener la solución óptima. La salida del programa también cambia. ]

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.

Puesto que los valores que representan tiempos pertenecen a un dominio continuo asumiremos que el esquema Programación dinámica no se puede aplicar. Por otra parte, el método voraz no garantiza la solución óptima ante cualquier instancia de este problema. **En esta práctica se pide aplicar una estrategia de vuelta atrás (*Backtracking*)** para encontrar la máxima ganancia que puede obtener el alfarero.

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, lo que produciría una ganancia de valor 30 cuya fabricación consumiría 10.4 unidades de tiempo.

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

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

**potter\_bt -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.

El programa mostrará únicamente tres resultados, cada uno en una línea distinta:

- Línea 1: Ganancia que el alfarero obtendría con la mejor planificación posible.
- Línea 2: Una estadística sobre los nodos del árbol de búsqueda de backtracking que consiste en los siguientes cinco valores:<sup>1</sup> (1) Número de nodos visitados; (2) número de nodos explorados; (3) número de nodos hoja que se visitan; (4) número de nodos descartados por no ser factibles y (5) número de nodos descartados por no ser prometedores. El carácter separador debe ser un espacio en blanco y deben mostrarse en ese orden.
- Línea 3: 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.

Por ejemplo:

```
potter_bt -f potter_n04r.def
30
81 9 1 62 9
0.058
```

## 3. Sobre la evaluación de esta práctica.

Un aspecto importante en la evaluación de este trabajo es el tiempo de respuesta. Trata de escribir un programa eficiente. Es muy recomendable que en el `makefile` incluyas la opción del compilador `-O3` (optimizador de código).

Se probarán todas las entregas con problemas de distintos tamaños y se obtendrán los tiempos de respuesta. En cada una de las pruebas de ejecución que se hagan, la práctica más rápida obtendrá la máxima calificación y la nota de todas las demás entregas se calculará de manera proporcional.

Aún así, cualquier trabajo tendrá la calificación de aprobado (en este apartado de evaluación) si obtiene una solución correcta ante problemas de pequeño tamaño, donde el tiempo de respuesta no es tan relevante.

---

<sup>1</sup>La magnitud de cada uno de estos valores depende de cada implementación particular por lo que no tienen por qué coincidir con los publicados (de hecho, lo normal es que no coincidan), incluso algunos de ellos podrían ser 0.

#### 4. 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_bt.cc`.
- b) Se debe entregar únicamente los ficheros `potter_bt.cc` y `makefile` (para generar el archivo ejecutable). 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*.<sup>2</sup> 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.

---

<sup>2</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>).