Algorítmica

Grado en Ingeniería Informática

Práctica 3

Formas de sumar n

Autores

María Jesús López Salmerón

Nazaret Román Guerrero

Laura Hernández Muñoz

José Baena Cobos

Carlos Sánchez Páez

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2017-2018

Índice

Índice de figuras

1. Descripción de la práctica

El objetivo de esta práctica es calcular todas las maneras posibles de sumar un número n utilizando los elementos pertenecientes a un conjunto $C = \{1, ..., n\}$.

2. Algoritmos implementados

Hemos implementado tres algortimos diferentes: uno con fuerza bruta y dos con bactracking. Estos dos últimos se diferencian en la manera de almacenar la información referente a los cálculos llevados a cabo.

2.1. Algoritmo de fuerza bruta

En este algoritmo evaluamos todas las posibles combinaciones de números, comenzando por el primer elemento del conjunto; este primer elemento se va sumando con los elementos sucesivos, comprobando si dicha suma alcanza nuestro objetivo n.

2.2. Algoritmos backtracking

- **Solución parcial**: tupla de tamaño fijo que contiene el valor 1 en el caso de que el elemento correspondiente a dicha posición se encuentre dentro de la solución y 0 en otro caso.
- **Restricciones explícitas**: el conjunto debe estar ordenado en orden no decreciente.
- **Restricciones implícitas**: la suma resultante de cada tupla debe ser igual a n y no debe haber dos elementos repetidos.

2.2.1. Algoritmo backtracking sin información

- **Función de factibilidad**: se comprueba si al añadir el siguente elemento a la suma no sobrepasamos n, si al sumar los elementos que ya tenemos y los restantes somos capaces de llegar a n y por último si la solución parcial es, en efecto, una solución.

2.2.2. Algoritmo backtracking con información

- **Función de factibilidad**: es la misma que la del caso anterior con una diferencia: en esta versión la suma actual y la suma de los elementos restantes se almacenan en variables de forma que no haya que calcularlas en cada iteración. No obstante, hay que tener una precaución: debemos resetear ambas variables cada vez que encontramos una solución.

3. Análisis de eficiencia

	Tamaño inicial	Tamaño final	Número de ejecuciones
Fuerza bruta Backtracking sin información Backtracking con información	1	25	24

Figura 1: Tamaños utilizados para la ejecución

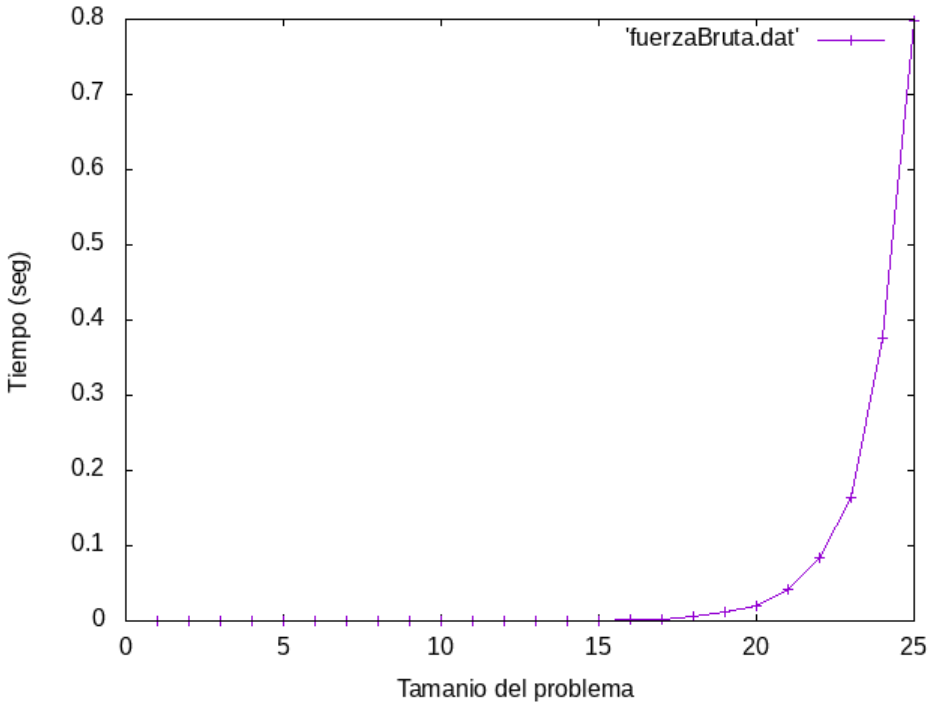

Figura 2: Fuerza bruta

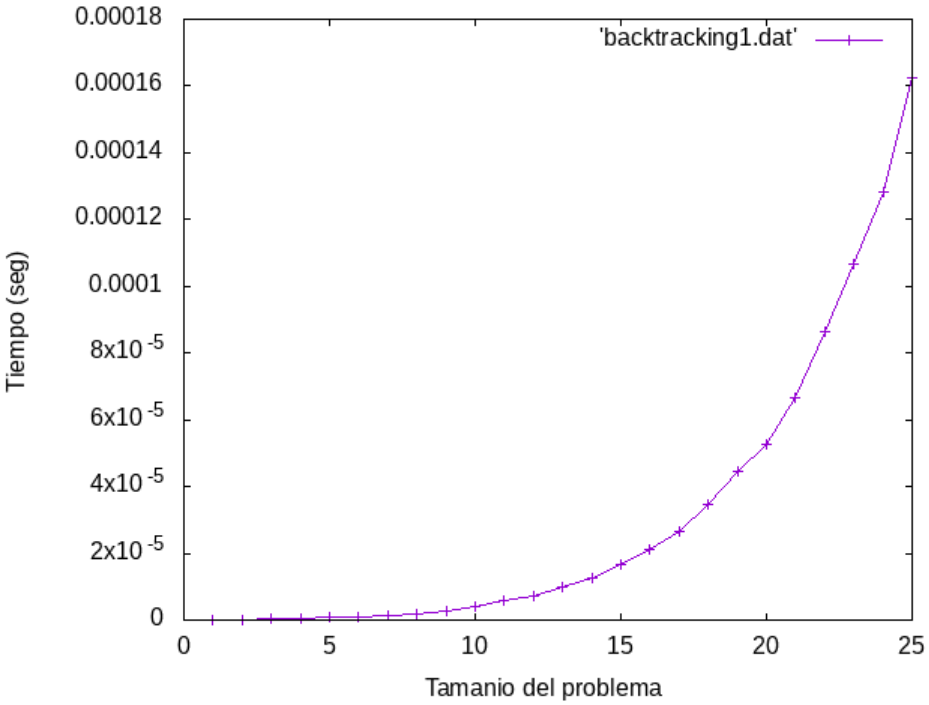

Figura 3: Backtracking sin información

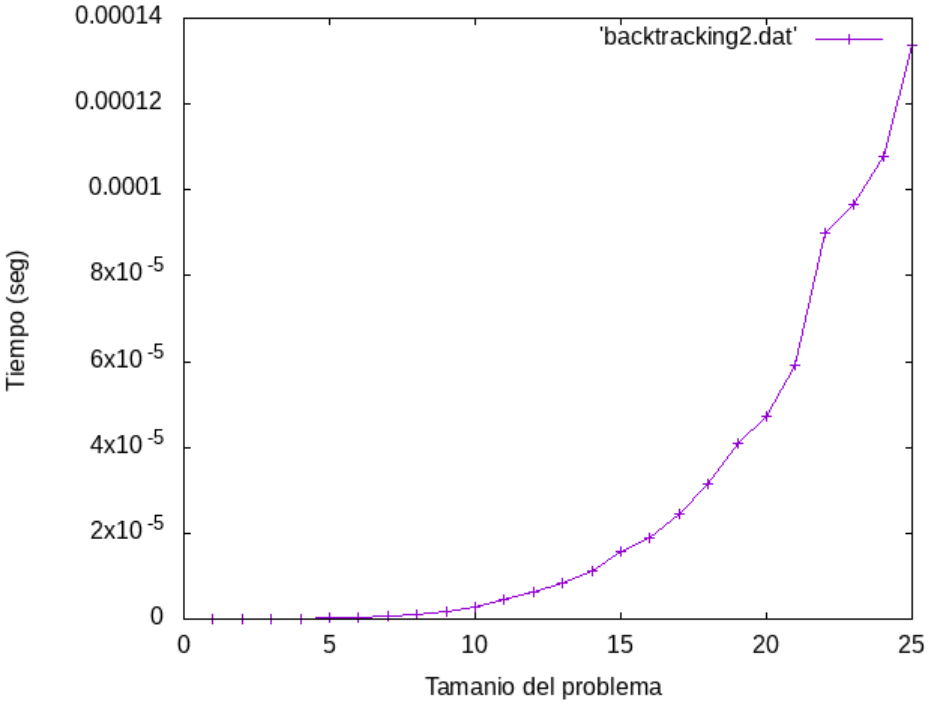

Figura 4: Backtracking con información

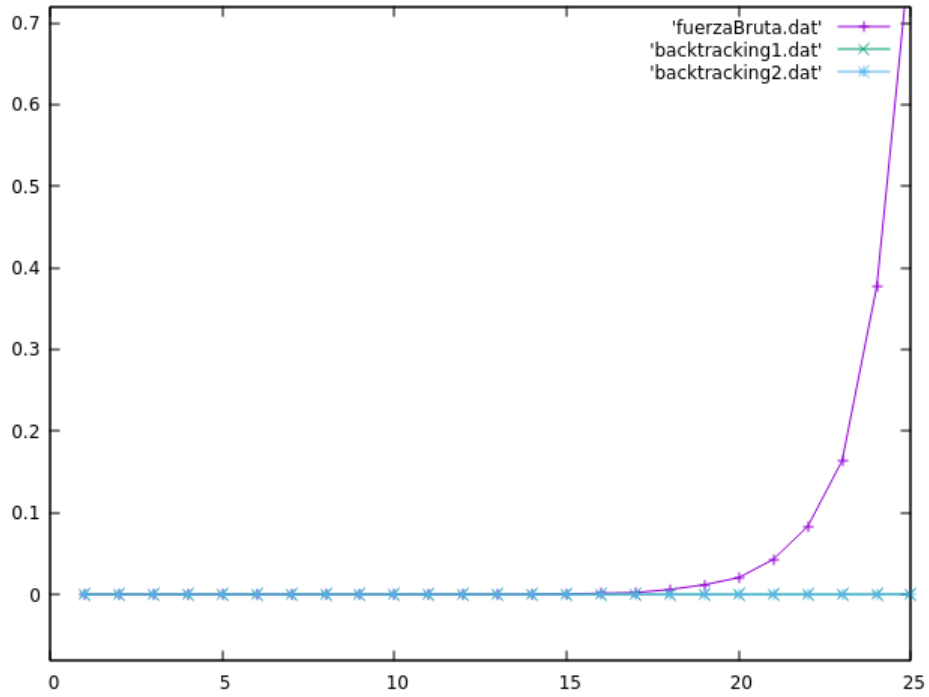

Figura 5: Comparativas entre algoritmos

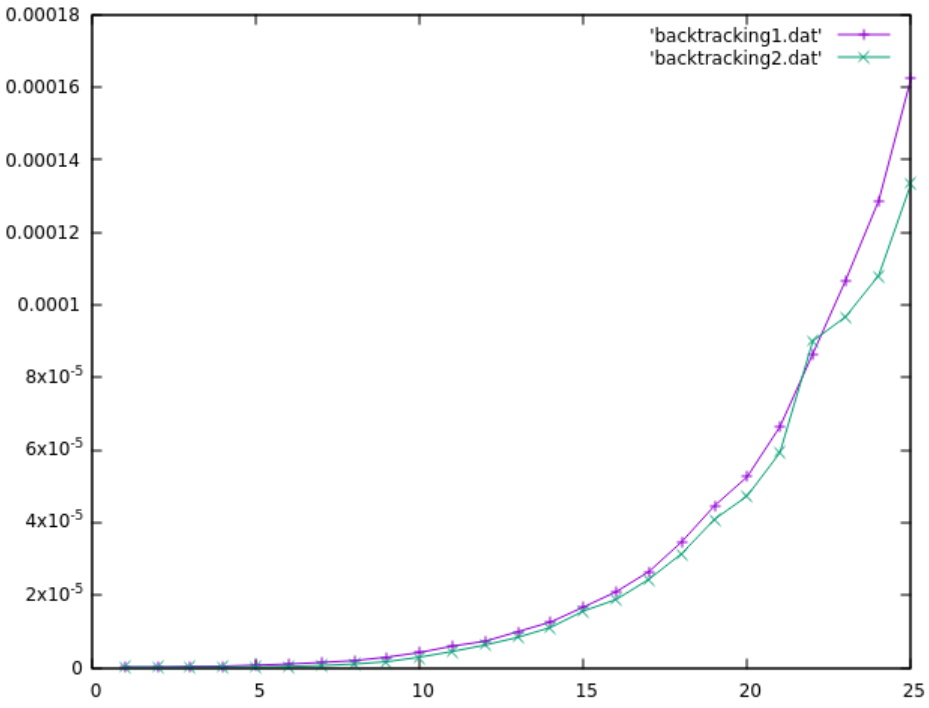

Figura 6: Comparativa entre algoritmos backtracking

4. Conclusiones

Como hemos podido observar en las figuras anteriores, el algortimo fuerza bruta es nefasto. La eficiencia de este algoritmo es factorial lo cual da lugar a tiempos muy elevados.

Sin embargo con backtracking, conseguimos órdenes de eficiencia mucho menores por el simple hecho de realizar una comprobación de factibilidad.

Pero podemos afinar aún más; si almacenamos los datos que se evalúan en la función de factibilidad evitamos tener que calcularlos en cada iteración, transformando una función de orden de eficiencia lineal en una función de orden constante.