

BACKTRACKING Y FUERZA BRUTA

El problema de la suma de Subconjuntos: Eficiencia

Álvaro Fernández García

En este documento se pretende ilustrar una comparativa acerca de la eficiencia de tres algoritmos disitintos para resolver el problema de la suma de Subconjuntos.

Dados $n+1$ números positivos: w_i , $1 \leq i \leq n$, y uno más M , se trata de encontrar todos los subconjuntos de números w_i cuya suma valga M .

Los algoritmos planteados son los siguientes:

- Un algoritmo de fuerza bruta.
- Un algoritmo de backtracking con una función de factibilidad ineficiente.
- Un algoritmo de backtracking con una función de factibilidad eficiente.

El algoritmo de fuerza bruta empleado es el siguiente:

```
/* REPRESENTACIÓN DE LAS TUPLAS: Vector de 0 o 1, 1 indica que se
selecciona dicho número, 0 que no se selecciona. */

void ProcesaVector(vector<int> & Tupla, vector<int> subconjunto, int sumaBuscada){
    int suma = 0;

    for(int i = 0; i<subconjunto.size(); ++i)
        if(Tupla[i] == 1)
            suma+=subconjunto[i];

    if(suma == sumaBuscada){
        for(int i = 0; i<subconjunto.size(); ++i)
            if(Tupla[i] == 1)
                cout << subconjunto[i] << " ";
            cout << endl;
    }
}

void completa_binario(vector<int> & V, vector<int> subconj, int pos, int sumaBuscada){

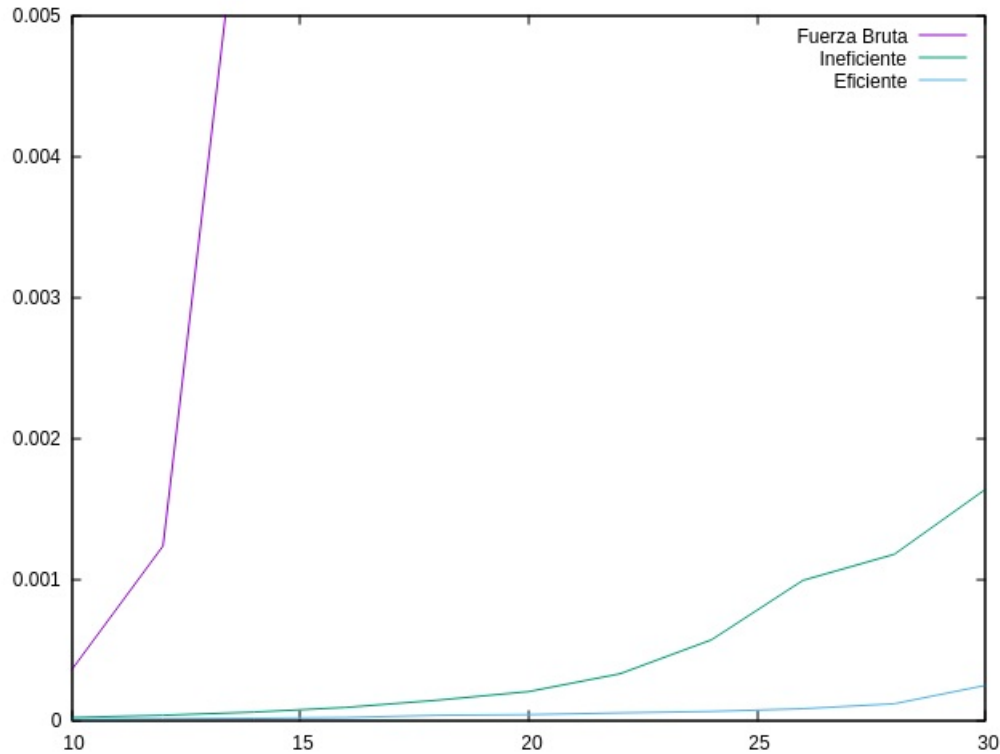
    if(pos==V.size())
        ProcesaVector(V, subconj, sumaBuscada);
    else{
        V[pos]=0;
        completa_binario(V,subconj, pos+1, sumaBuscada);
        V[pos]=1;
        completa_binario(V,subconj, pos+1, sumaBuscada);
    }
}

void SumaConjuntosFuerzaBruta(vector<int> subconjunto, int sumaBuscada){
    vector<int> tuplaIni;
    tuplaIni.resize(subconjunto.size());
    completa_binario(tuplaIni, subconjunto, 0, sumaBuscada);
}
```

```
}
```

Los dos algoritmos de backtracking tienen la misma implementación que la mostrada en las transparencias de clase.

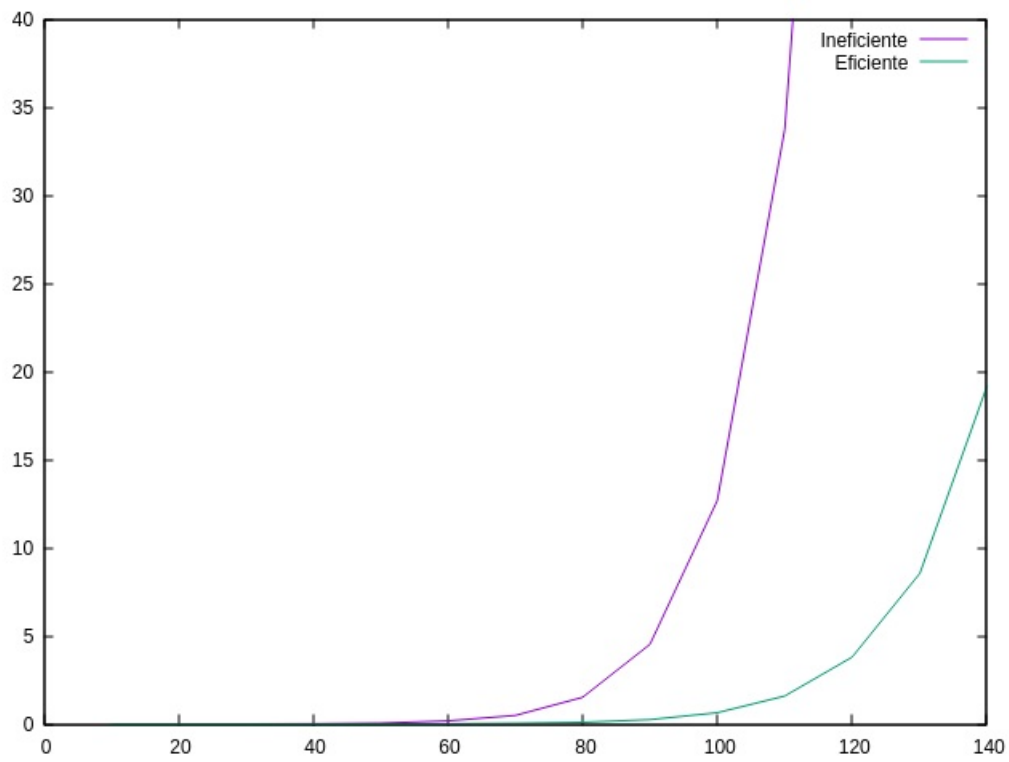
El estudio empírico de la eficiencia de los algoritmos muestra la siguiente gráfica comparativa:



Las mediciones se han realizado para vectores de tamaño 10 a 30 variando su tamaño de dos en dos (*ya que para tamaños superiores, el algoritmo de Fuerza Bruta se volvía impracticable*) en las que el valor de la suma buscado es el elemento enésimo multiplicado por 2 para hacer el caso lo más homogéneo posible en todas las mediciones.

Como conclusión podemos ver que los algoritmos de backtracking superan con creces la eficiencia de los algoritmos de Fuerza Bruta.

A continuación se muestra una gráfica comparativa más precisa entre ambos algoritmos de backtracking. Las mediciones se han realizado de 10 a 140 variando el tamaño de 10 en 10 y con el mismo criterio que antes para la suma buscada.



Como conclusión obtenemos que en el diseño de este tipo de algoritmos influye bastante la elección de una función de factibilidad que sea lo más eficiente posible.