

PROGRAMACIÓN III

INFORME DE LA PRÁCTICA 2 (FUERZA BRUTA/GREEDY)

Nombre de los participantes:

- Marcos Jesús Santana Pérez
- Chetani Mesa Guzmán
- Ían Marrero Martín

Descripción del problema a resolver:

- Por medio de combinaciones sin repetición, donde si importa el orden y los elementos son cogidos de n en n , queremos obtener el mayor valor generado de entre todas las permutaciones.

Lenguajes de programación escogidos:

- JAVA v1.8.0_191
- PYTHON v2.7
- C

Estrategia de Fuerza Bruta

1. Nombre de los ficheros que contienen el algoritmo de fuerza bruta implementado

- JAVA => [Pr3/Java/src/Max](#)/Main.java
- PYTHON => [Pr3/Python](#)/main.py
- C =>

2. Copia de pantalla del fragmento de código que implementa el algoritmo de fuerza bruta (en los 3 lenguajes).

- JAVA

Nota: en java usamos BigInteger como portador de la concatenación de valores del array elementos (en el orden indicado por el array indices), puesto que el valor generado por dicha concatenación puede ser muy grande, incluso superior al que puede contener un long.

```
public BigInteger Iterador (String [] elementos){
    int[] indices; // orden a mostrar de elementos
    Permutator p = new Permutator (elementos.length);
    BigInteger maximo = BigInteger.ZERO;
    while (p.hasMore ()) {
        indices = p.getNext (); //obtenemos siguiente permutacion
        BigInteger n = lstToBigInteger(elementos, indices);
        if(n.compareTo(maximo) > 0){ // comprobamos si el nuevo n es mayor que maximo
            maximo = n;
        }
    }
    return maximo;
}
```

- PYTHON

```
# maximo contiene el valor mas grande encontrado hasta el momento
maximo = 0
# Hacemos una primera medida del time clock
time.clock()
for permutacion in itertools.permutations(lst):
    # Construimos el numero a partir de la lista permutada
    n = etc.lstToInt(permutacion)
    # Comprobamos si el nuevo n es mayor que maximo
    if n > maximo:
        maximo = n
```

- C

3. Nombre de los ficheros que contienen el iterador (sólo en los lenguajes en los que hayan tenido que implementar el iterador)

- JAVA => [Pr3/java/src/Itertools](#)/Permutator.java
- C =>

4. Explicación del iterador de permutaciones o combinaciones utilizado en cada lenguaje (en caso de haber utilizado un iterador disponible en el lenguaje, referenciar la página Web de documentación del iterador utilizado; en caso de haberlo programado, referenciar la página Web que describe el algoritmo implementado, o explicar el algoritmo implementado).

- JAVA => http://people.rennes.inria.fr/Arnaud.Gotlieb/resources/java_exp/min/perm.htm
- PYTHON =>
- C => Mismo algoritmo implementado en java, pero adaptado a C.

5. Formato del fichero de entrada del programa

Formato ".txt"

6. Copia de pantalla que muestre el uso del programa desde consola activando la opción que muestra el tiempo consumido en la ejecución del programa (en los tres lenguajes)

- JAVA =>
- PYTHON =>
- C =>

Información adicional