

# Técnicas de diseño de algoritmos

## Búsqueda exhaustiva

### Ejercicios (Bloque 2): con pautas de resolución

Luis Javier Rodríguez Fuentes  
Amparo Varona Fernández

Departamento de Electricidad y Electrónica  
Facultad de Ciencia y Tecnología, UPV/EHU

[luisjavier.rodriguez@ehu.es](mailto:luisjavier.rodriguez@ehu.es)

[amparo.varona@ehu.es](mailto:amparo.varona@ehu.es)

OpenCourseWare 2015  
Campus Virtual UPV/EHU

## Búsqueda exhaustiva – Ejercicios (Bloque 2)

- (B2.1) Considerese una lista  $s$  con  $n$  letras, todas ellas diferentes. Escribir en lenguaje Python una función que retorne la lista de todas las palabras formadas por  $m \leq n$  letras distintas que puedan construirse con ellas, usando un esquema de búsqueda exhaustiva.

Lo que se pide es un problema clásico de combinatoria: encontrar las variaciones de  $n$  elementos tomados de  $m$  en  $m$ . La función deberá obtener el conjunto de soluciones que se derivan de un prefijo  $p$  de longitud  $k$  (donde todas las letras son diferentes), creando un nuevo prefijo  $q$  con cada una de las letras no presentes en  $p$  y añadiendo al conjunto solución de  $p$  todas las palabras cuyo prefijo es  $q$ . La función retornará el conjunto solución. Si el prefijo  $p$  tiene la longitud deseada  $m$ , el conjunto solución estará formado por el propio prefijo  $p$ .

## Búsqueda exhaustiva – Ejercicios (Bloque 2)

(B2.2) Sean dos grafos dirigidos  $g_1$  y  $g_2$ , ambos con  $n$  nodos. Se dice que  $g_1$  y  $g_2$  son *isomorfos* si existe una función biyectiva  $f : V(g_1) \rightarrow V(g_2)$  entre los vértices de ambos grafos tal que  $\langle x, y \rangle \in E(g_1)$  si y sólo si  $\langle f(x), f(y) \rangle \in E(g_2)$ . Escribir en lenguaje Python un algoritmo que, aplicando el esquema de búsqueda exhaustiva, determine si dos grafos  $g_1$  y  $g_2$  son isomorfos.

La función retornará `True` si los grafos son isomorfos y `False` en caso contrario. Suponiendo que los vértices de ambos grafos se identifican implícitamente mediante los enteros  $1, 2, \dots, n$ , la función  $f$  buscada (en caso de que exista) será una permutación de los vértices del grafo  $g_1$  que los relaciona con los vértices del grafo  $g_2$ :  $(\sigma(1), \sigma(2), \dots, \sigma(n))$ . Así pues, la búsqueda consistirá en recorrer estas permutaciones y comprobar si coinciden los arcos de entrada y salida que conectan el vértice considerado con los vértices ya presentes en la solución parcial. La función sólo explorará soluciones parciales donde no haya incongruencias entre  $g_1$  y  $g_2$ . La función retornará `True` sólo si alcanza una solución parcial de longitud  $n$ .