

## Laboratory practice No. 5: Divide and conquer and Dynamic programming

**Santiago Albisser**

Universidad EAFIT  
Medellin, Colombia  
salbisserc@eafit.edu.co

**Juan Pablo Leal**

Universidad EAFIT  
Medellin, Colombia  
jplealj@eafit.edu.co

Mayo 5, 2019

3.1) Para el algoritmo se usaron dos matrices: una que contiene los costos de los caminos y otra con los vértices visitados. El algoritmo calcula el costo que hay desde el primer vértice hasta cada uno de los otros y los va almacenando. En la última iteración se obtienen posibles caminos y se escoge el de mínimo costo.

3.2)  $O(2^n)$

3.3) Se usó un grafo como una matriz de adyacencia en la que se guardan las posiciones de cada pila de basura y de la posición inicial. La distancia se podría hallar con el algoritmo de Held-Karp.

3.4) Se usó un grafo como una matriz de adyacencia en la que se guardan las posiciones de cada pila de basura y de la posición inicial.

3.5)  $O(n^2 * 2^n)$

3.6) N es el tamaño de la matriz.

4.2.1)  $O(\text{lenx} * \text{leny})$

4.2.2) table [ lenx] [leny]

4.3.1) a.

4.3.2)  $C1:n + c2$

4.4.1) c

4.5) c

4.5.2) a[mitad]

4.5.3) a, mitad+1, de, z

4.6.1)  $\text{scm}[i] = 1;$

4.6.2)  $\text{scm}[i] = 1 + \text{scm}[j]$

4.6.3)  $\text{max} = \text{scm}[i];$

4.6.4) c

4.7.1)  $d[i][j];$

4.7.2)  $d[k][j];$

4.7.3)  $d[i] [k];$

4.7.4)  $O(n^3);$