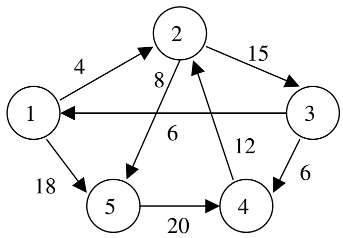
Algorítmica
Hoja 4 de problemas – Programación dinámica

1. Resolver por programación dinámica el problema del camino mínimo (dando explicitamente cada camino) entre cada par de vértices para el grafo siguiente:



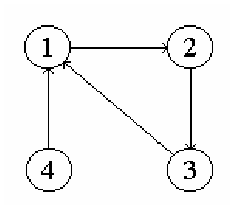
2. Modificar el pseudocódigo del algoritmo Caminos_mínimos_dinámico sustituyendo la línea:

   A[i,j] ← min(A[i,j],A[i,k]+A[k,j])

   por una secuencia de instrucciones condicionales que lleve la cuenta del cálculo del camino mínimo en una matriz auxiliar C.

3. Sea un grafo dirigido G = (V,A) con A implementada como matriz de adyacencia. La clausura transitiva reflexiva de A es la matriz A* con la propiedad de que A*(i,j) = cierto si y sólo si G posee un camino dirigido de i a j.

   (a) Calcular la matriz A* para el grafo siguiente:



   (b) Sea $A^k(i,j) = 1$ si y sólo si existe un camino dirigido entre i y j que pase por vértices de número menor o igual a k como mucho. Definir A^0 en términos de la matriz G.A.

   (c) Obtener una ley de recurrencia que permita obtener A^k a partir de A^{k-1}, usando los operadores lógicos o é y.

   (d) Escribir un algoritmo iterativo que calcule A* con memoria en el orden de n^2.

4. Proponer un ejemplo de objetos en el Problema de la Mochila de modo que se cumpla que $|f^k| = 2^k$ para todos los k=0,...,n.

5. Resolver el problema de optimización de la suma de subconjuntos por programación dinámica. El problema se plantea de la forma siguiente: Dado un conjunto S de *n* números enteros positivos, y un número entero positivo C, encontrar el subconjunto de S de menor cardinal tal que la suma de sus elementos sea C, si existe tal conjunto. Para resolver este problema por programación dinámica, se pide:

   (a) Plantear un ejemplo de conjunto S y de cantidad C que demuestren que la estrategia de avance rápido consistente en elegir los elementos de S por orden descendente no lleva necesariamente a la solución óptima.

(b) Plantear el problema como una secuencia de *n* decisiones. Escribir cuales son las opciones de la decisión i-ésima. Plantear el principio del óptimo para este problema y demostrarlo.

(c) Se define $f_i(X)$ como el número óptimo de elementos de S para sumar la cantidad X, usando sólo los elementos desde el primero hasta el i-ésimo. Escribir la fórmula que permite calcular $f_i(X)$ en términos de $f_{i-1}()$, con especial atención a los casos límite. ¿Qué valores de i y de X se deben tomar para que $f_i(X)$ sea el valor de la función objetivo en la solución del problema?

(d) La función $f_i(X)$ se puede calcular usando una matriz adecuada, y rellenándola usando la fórmula obtenida en el apartado anterior. Escribir el pseudocódigo de un algoritmo que construya la tabla, la rellene y la devuelva. ¿Cuál es su complejidad?

(e) A partir de la tabla del apartado anterior, es posible reconstruir la solución del problema. Escribir el pseudocódigo de un algoritmo que reciba como entrada la tabla del apartado anterior y que devuelva una matriz de n elementos que valga cierto para los índices de los elementos que están en el subconjunto solución, y falso para los demás. ¿Cuál es su complejidad?