

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 6

Juan Navarro

Problema 1

Se dice que un grafo no dirigido $G = (V, E)$ es *simple* si para todo arco $(x, y) \in E$, se tiene que $x \neq y$, y además no se permite más de un arco entre dos nodos.

Dada una lista de n enteros positivos $d_1, \dots, d_n$, se busca determinar si existe un grafo simple $G = (V, E)$, con nodos $V = \{v_1, \dots, v_n\}$, tal que el grado del nodo v_i sea exactamente igual a d_i .

Parte 1 Muestre que no es suficiente verificar que la suma de los grados sea par. Es decir que existe una secuencia de enteros, donde la suma de sus elementos es par, pero donde no existe un grafo simple que la satisfaga.

Idea de solución 3,3,1,1

Parte 2 Suponiendo que $d_1 \geq d_2 \geq \dots \geq d_n$, y que existe un grafo $G = (V, E)$ con secuencia $d_1, \dots, d_n$, busquemos demostrar que existe un grafo con la misma secuencia, donde además los vecinos de v_1 son $\{v_2, \dots, v_{d_1+1}\}$.

1. Suponiendo que los vecinos de v_1 no son $v_2, \dots, v_{d_1+1}$, muestre que existen $i < j \leq n$ y $u \in V$ tales que $(v_1, v_i), (u, v_j) \notin E$, y $(v_1, v_j), (u, v_i) \in E$.

Idea de solución Por enunciado, existe un elemento entre $v_2, \dots, v_{d_1+1}$ que no está conectado con v_1 , con el cual se puede definir i . Además existe algún valor mayor a $d_1 + 1$ tal que $(v_1, v_j) \in E$, el cual corresponde a j . Por otra parte, como todos los valores de d son positivos, $d_i \geq d_j$, $(d_j, d_1) \in E$, y $(d_i, d_1) \notin E$, entonces debe existir algún nodo u conectado con v_i que no esté conectado con v_j .

2. Muestre como modificar G para obtener G' con la misma secuencia, y donde $(v_1, v_i) \in E'$.

Solución Eliminar los arcos $(v_1, v_j), (v_i, u)$ y agregar los arcos (v_1, v_i) y (v_j, u) .

3. Muestre que debe existir un grafo con la misma secuencia, pero donde los vecinos de v_1 son $v_2, \dots, v_{d_1+1}$.

Solución El procedimiento anterior se puede aplicar mientras se cumplan las condiciones, y además tras cada aplicación, el valor de

$$\sum \{i | (v_1, v_i) \in E\}$$

disminuye.

Luego, como la única forma en que las condiciones no se cumplan es que los vecinos de v_1 sean $v_2, \dots, v_{d_1+1}$, y esto sucederá tras una cantidad finita de usos del procedimiento, entonces existe un grafo con la propiedad buscada.

Parte 3 Describa un algoritmo que con input $d_1, \dots, d_n$ (posiblemente desordenados), decida si existe un grafo simple con dicha secuencia de grados.

Solución

```

function DEGREESEQUENCES( $d = (d_1, \dots, d_n)$ )
  Ordenar  $d$  en orden no creciente
     $\triangleright$  También se puede ordenar la primera vez, y después mezclar las dos partes ordenadas
  if  $d$  es vacía, o  $d_1 = 0$  then
    return True
  end if
  if  $d_1 \geq n$  then
    return False
  end if
  for  $i = 2..(d_1 + 1)$  do
     $d[i] \leftarrow d[i] - 1$ 
    if  $d[i] < 0$  then
      return False
    end if
  end for
  return DEGREESEQUENCES( $[d_2, \dots, d_n]$ )
end function

```

Problema 3

Una secuencia es un *palíndromo* si es igual a su reverso. Describa y analice un algoritmo que dada una secuencia $\{d_1, \dots, d_n\}$, determine el largo de la subsecuencia más larga que sea un palíndromo.

Idea de solución La subsecuencia entre (i, j) es un palíndromo si y solo si la secuencia entre $(i + 1, j - 1)$ también lo es, y los elementos en las posiciones i y j son iguales. Lo anterior permite la definición:

$$Pal(i, j) = \begin{cases} 1 & i \geq j \\ 1 & d_i = d_j \text{ y } Pal(i + 1, j - 1) \\ 0 & \text{en otro caso} \end{cases}$$

que puede calcularse con programación dinámica en tiempo $O(n^2)$. El valor a retornar se puede buscar en la tabla, revisando las diagonales que correspondan, o bien se puede mantener y actualizar durante la ejecución del algoritmo.

Problema 4

Se busca dividir un string de largo n en $m + 1$ substrings, con cortes en las posiciones $\{p_1, \dots, p_m\}$. Para esto, se dispone de una operación que recibe un string y una posición, y entrega los dos subtrings correspondientes. Por la forma en que está implementada, esta operación copia el string recibido, por lo que su costo es igual al largo del string.

Describa y analice un algoritmo que determine el costo mínimo de cortar el string en las $m + 1$ partes. Se consideran como operaciones básicas las comparaciones entre dos elementos, y las operaciones de suma y multiplicación.

Idea de solución Como notación, dadas dos posiciones i y j entre 0 y n , se define

$$P(i, j) = \{p \in P \mid p > i \text{ y } p < j\}.$$

Luego, si denominamos por m_{ij} al tamaño de $P(i, j)$, entonces el costo mínimo de realizar m_{ij} cortes entre las posiciones i y j está dado por:

$$C(i, j) = \begin{cases} 0 & m_{ij} = 0 \\ (j - i) + \min_{p \in P(i, j)} (C(i, p) + C(p, j)) & m_{ij} > 0 \end{cases}$$

que se puede calcular en tiempo $O(m^3)$ usando programación dinámica. El orden de los cortes se puede recuperar modificando el algoritmo.