CC3101 — Matemáticas Discretas para Computación

Soluciones Control 3

12 de enero de 2012

1. Un *bosque* es un grafo simple sin ciclos. Conteste cada una de las siguientes preguntas justificando claramente en cada caso.

   a) ¿Cuál es la diferencia entre un bosque y un árbol?

   b) Si F es un bosque con n vértices y m componentes conexas ¿Cuántas aristas en total tiene F?

   c) ¿Cuál es el mínimo número de aristas que puede tener un bosque de n vértices?

   d) ¿Cuál es el máximo número de aristas que puede tener un bosque de n vértices?

   Solucion:

   a) Un bosque no necesariamente es conexo, pero cada componente conexa es un árbol (un bosque se puede ver como un conjunto de árboles)

   b) Cada componente conexa es un árbol, luego cada componente conexa tiene exactamente una arista menos que la cantidad de vértices de la componente. Si suponemos que las componentes conexas de F tienen $n_1, n_2, \dots, n_m$ vértices cada una entonces la cantidad total de aristas de F es $(n_1 - 1) + (n_2 - 1) + \cdots + (n_m - 1) = (n_1 + n_2 + \cdots + n_m) - m = n - m$. Esto demuestra que la cantidad de aristas del bosque es $n - m$.

   c) El mínimo se alcanza cuando hay tantas componentes como vértices, es decir cuando el bosque está compuesto de sólo vértices aislados. En ese caso la cantidad de aristas es 0.

   d) La máxima cantidad de aristas se alcanza cuando el bosque tiene una componente conexa, es decir cuando es un árbol, en cuyo caso tiene exactamente $n - 1$ aristas. Sabemos que no puede tener más que esas aristas porque de otra forma el grafo dejaría de ser un árbol y por lo tanto tendría un ciclo y no sería un bosque.

2. Responda "verdadero" o "falso" para cada una de las siguientes afirmaciones. En cada caso demuestre formalmente su respuesta.

   a) Suponga que G es un grafo Euleriano en que las aristas e y f tienen un extremo común. Entonces G tiene un ciclo Euleriano en que e y f aparecen en forma consecutiva.

   b) Sea G un grafo conexo tal que todos sus vértices de grado mayor que 1 son vértices de corte. Entonces G es un árbol.

   c) Un grafo se dice par si todos su vértices tienen grado par. Sea G un grafo par con k componentes conexas y e una arista cualquiera de G. Entonces $G - e$ también tiene k componentes conexas.

   Solución:

   a) Falso. Considere el grafo G con aristas $E(G) = \{uv, uw, wv, vx, xy, vy\}$. Este grafo es Euleriano (conexo y todos los vértices de grado par). Considere las aristas $e = uv$ y $f = wv$. Ambas tienen un extremo en común pero se puede demostrar que no hay un ciclo Euleriano en que aparezcan consecutivas. Por contradicción: suponga que hay un ciclo en G en que aparece e seguido de f, es decir el ciclo tiene una porción de la forma (u, v, w). Note que la única arista distinta de f que sale de w va hacia el vértice u, y similarmente, la única arista distinta de e que sale de u va a w, luego el único ciclo en G que tiene a e seguido de f es el ciclo (u, v, w, u) que no es un ciclo Euleriano en G.

b) Falso. Considere el grado G con aristas $E(G) = \{u_1v_1, v_1v_2, u_2v_2, v_2v_3, u_3v_3, v_3v_1\}$. Note que los únicos vértices de grado 1 son u_1, u_2 y u_3. Más aun, v_1, v_2 y v_3 son vértices de corte (sacarlos aísla respectivamente a u_1, u_2 o u_3). Luego todos los vértices de G que tienen grado mayor a 1 son de corte, pero G tiene un ciclo (v_1, v_2, v_3, v_1) luego G no es un árbol.

c) Verdadero. Dado que todos los vértices de G tienen grado par, cada componente conexa de G tiene un ciclo que pasa por todas sus aristas (cada componente por separado es un grafo Euleriano, a pesar de que G no es necesariamente Euleriano ya que no es conexo). Luego todas las aristas de G pertenecen a almenos un cilo y por lo tanto ninguna arista es de corte. En particular, la arista e no es de corte lo que implica que la cantidad de componentes conexas de G y de $G - e$ es la misma e igual a k.

3. El siguiente algoritmo iterativo se usa para buscar el valor k dentro de la secuencia ordenada crecientemente $S = (s_1, s_2, \ldots, s_n)$.

INPUT: Una secuencia de enteros $S = (s_1, s_2, \ldots, s_n)$ tal que $s_1 \leq s_2 \leq \cdots \leq s_n$, un natural n correspondiente al largo de la secuencia y un número entero k que pertenece a la secuencia.
OUTPUT: Un índice i tal que $s_i = k$.

```
Bin-Search(S, n, k)
a := 1
b := n
m := ⌊(a + b)/2⌋
while b − a > 0
    if k > s_m then
        a := m + 1
    if k ≤ s_m then
        b := m
    m := ⌊(a + b)/2⌋
return a
```

Demuestre que el algoritmo es correcto.

Solución (borrador): Se puede demostrar que el siguiente es un invariante para el único loop del algoritmo:

Después de cada iteración se tiene que $a \leq b$
y el valor k se encuentra en la secuencia $s_a, s_{a+1}, \ldots, s_b$.

Dado el invariante, es fácil demostrar que si el algoritmo se detiene entonces $b - a \leq 0$ y dado que $a \leq b$ se tiene que $a = b$, luego $k = s_a$ y por lo tanto a es el índice buscado. Para justificar terminación basta con argumentar que el valor de $b - a$ es un natural que siempre decrece, por lo tanto en algún momento será menor o igual a 0 y el loop terminará.