Pontificia Universidad Catolica de Chile
DCC
Semestre 2018-2

Curso: IIC2133 - EDD
Profesor: Yadran Eterovic

Ayudantía

Repaso I3

7 de noviembre de 2018

1. Supongamos que G es no direccional (las aristas no tienen dirección). Decimos que G es **biconectado** si no hay ningún vértice que al ser sacado de G desconecta el resto del grafo. Por el contrario, si G no es biconectado, entonces los vértices que al ser sacados de G desconectan el grafo se conocen como **puntos de articulación**.

   Describe un algoritmo eficiente para encontrar todos los puntos de articulación en un grafo conectado. Explica cuál es la complejidad de tu algoritmo.

2. Un árbol de extensión de costo mínimo incluye para cada vértice v una arista de costo mínimo incidente en v. Demuestre o refute esta afirmación.

3. El diámetro de un grafo direccional con costos es la más larga de las rutas más cortas en el grafo. En clase, estudiamos el algoritmo de Floyd-Warshall para calcular la *longitud* de las rutas más cortas entre todos los pares de nodos de un grafo. Da un algoritmo eficiente, por ejemplo, basado en el algoritmo de Floyd-Warshall, que imprima la *secuencia de vértices* del diámetro de un grafo.

4. Considera el algoritmo de Prim para encontrar un árbol de cobertura de costo mínimo para un grafo no direccional `G = (V, E)`, a partir del vértice `r` en `E`:

```
for ( cada vértice u en V ) u.key = ∞
r.key = 0
formar una cola Q con todos los vértices en V, priorizada según el campo key de cada vértice
π[r] = null
while ( !Q.empty( ) )
    u = Q.extractMin( ) —esta operación modifica la cola Q
    for ( cada vértice v en listadeAdyacencias[u] )
        if ( v ∈ Q ∧ costo(u,v) < v.key )
            π[v] = u
            v.key = costo(u,v) —esta operación modifica la cola Q
```

   Recuerda que el desempeño de Prim depende de cómo se implementa la cola `Q`; p.ej., si `Q` es un heap bina- rio, entonces Prim toma tiempo `O(E logV)`: el for dentro del while mira cada arista de `G` y para cada una rea- liza una operación decreaseKey sobre un vértice (cuando actualiza `v.key`, en la última línea).

Si los costos de todas las aristas de G son números enteros entre 0 y una constante `W` (tal que es factible declarar un arreglo de tamaño `W+1`), describe una forma de implementar la cola `Q`, tal que Prim corra en tiempo `O(E)`, es decir, que cada operación descreaseKey tome tiempo `O(1)`. [Recuerda que si `W` es constan- te, es decir, no depende ni de `E` ni de `V`, entonces recorrer un arreglo de tamaño `W+1` toma tiempo `O(1)`.]

5. Podemos usar las discrepancias en las tasas de cambio de monedas para transformar una unidad de una moneda en más de una unidad de la misma moneda. Por ejemplo, si un euro compra 1.38 dólares, un dólar compra 500 pesos, y un peso compra 0.0015 euros, entonces, a partir de un euro podemos obtener $1{,}38 \cdot 500 \cdot 0{,}0015 = 1{,}035$ euros.

   Hay n monedas $\langle c_1, c_2, ..., c_n \rangle$ y una tabla R de $n \times n$ tasas de cambio: una unidad de la moneda c_j compra $R[j][k]$ unidades de la moneda c_k. Queremos determinar si existe o no una secuencia de monedas $\langle c_a, c_b, ..., c_m \rangle$ tal que:

$$R[a][b] \cdot R[b][c] \cdot ... \cdot R[m][a] > 1$$

   a) Muestra que este problema puede plantearse como un problema de determinar si en un grafo direccional existe un ciclo con costo total negativo. *[Recuerda que $x > 1 \Rightarrow \frac{1}{x} < 1$ y que $log(ab) = log(a) + log(b)$.]*