Auxiliar #1 - Cotas Inferiores

7 de septiembre de 2020 - Bernardo Subercaseaux

Problema 1. (★★) Alicia y Roberto deciden jugar al ahorcado. Alicia elige un entero $k \geq 1$ y dibuja k líneas horizontales en una pizarra, simbolizando las letras que le restan a Roberto por adivinar para formar una palabra válida en el idioma español. En cada ronda del juego Roberto dice una letra, y si la letra pertenece a la palabra que Alicia tiene en mente, entonces ella escribe en la pizarra todas sus ocurrencias. De lo contrario Roberto pierde una *vida*. Pruebe que Roberto necesita comenzar con al menos 11 vidas para ganar el juego.

Solución 1 . Alicia (quien juega como *adversario*) elige $k = 4$. Para eso Alicia mantendrá en mente una lista $L = \{$cara, cama, cana, caña, capa, cada, casa, caza, caja, cava, cala$\}$ de palabras en que puede estar pensando. Inicialmente $|L| = 11$. Si Roberto dice 'c', Alicia revela la primera letra, y si Roberto dice 'a', entonces Alicia revela la segunda y cuarta letra. Cada letra que diga Roberto distinta de 'a' o 'c' es rechazada, costándole una vida. Dado que cada letra que le cuesta una vida a Roberto descarta a lo más 1 palabra de L, este requiere perder al menos 10 vidas para reducir L a una lista de tamaño 1 y así poder adivinar.

En los siguientes problemas trabajaremos en el modelo en que el input es un grafo que recibimos a través de su matriz de adyacencia A, y contaremos la cantidad de accesos $A(u, v)$ efectuados. Llamaremos *consulta de adyacencia* a cada uno de estos accesos. Cada consulta de adyacencia $A(u, v)$ puede pensarse como preguntarle *¿Hay una arista de u a v?* a un adversario.

Problema 2. (★★) En un grafo dirigido de n nodos se define un *sumidero* como un nodo al que entran $n - 1$ aristas y del cual no sale ninguna. Pruebe que para determinar si un grafo G de n nodos tiene un sumidero se requieren $\Omega(n)$ consultas de adyacencia. ¿Es esta cota ajustada?

Solución 2 . Podemos probar algo más fuerte incluso: se requieren $\Omega(n)$ consultas para determinar si el primer nodo (según el orden de A, indexando desde 1) es un sumidero. Para verificar si el primer nodo es un sumidero debemos verificar que $A(1, v) = 1$ para todo $v \neq 1$, y que $A(v, 1) = 0$ para todo v. Consideremos el adversario $\mathcal{E}$ que responde a cada pregunta $A(u, v)$ de la siguiente manera:

$$\mathcal{E}(u, v) = \begin{cases} 1 & \text{si } v = 1, u \neq 1 \\ 0 & \text{si no} \end{cases}$$

Notemos que la información que da $\mathcal{E}$ siempre permite decir que los nodos con índice ≥ 2 no son sumideros, así que el grafo tendrá un sumidero si y solo si el nodo 1 es sumidero. Si se han hecho menos de $n - 1$ consultas a $\mathcal{E}$, entonces existe un $v \neq 1$ tal que $\mathcal{E}(1, v)$ no ha sido consultado. En este caso cualquier respuesta que de un algoritmo será incorrecta; Si el algoritmo dice que no hay un sumidero, entonces $\mathcal{E}$ responde que el nodo 1 es un sumidero, lo que es consistente con la información dada. Si el algoritmo dice que 1 es un sumidero, entonces $\mathcal{E}$ aprovecha el indice v no consultado, y responde que 1 no es un sumidero ya que $A(1, v) = 0$, lo que es consistente con la información dada.

Curiosamente esta cota es ajustada. Se puede determinar la existencia de un sumidero con $O(n)$ consultas de adyacencia. Para esto consideremos el algoritmo que comienza con una lista de *candidatos* $C = \{1, \ldots, n\}$, que inicialmente incluye a todos los nodos. La idea es mantener en C

los nodos que podrían ser sumideros. Notemos que si C llega en algún momento a tener tamaño 1, entonces hay un único nodo que podría ser sumidero, y solo restaría verificar que realmente lo sea. Mientras $|C| \geq 2$, consideramos dos elementos u, v cualesquiera de C y consultamos $A(u, v)$. Si hay arista de u a v entonces u no puede ser sumidero. Si no hay arista de u a v entonces v no puede ser sumidero. En cualquier caso hemos descartado un candidato y reducido el tamaño de C en 1. Así podemos llevar $|C|$ a 1 haciendo $n - 1$ preguntas. Llamemos c al único elemento en C al terminar el paso anterior. Finalmente usamos $O(n)$ preguntas adicionales para verificar que $A(c, u) = 1$ y $A(u, c) = 0$ para todo u.

Problema 3. (★★) Demuestre que determinar si un grafo no dirigido de n nodos tiene un nodo de grado 3 se requieren $\binom{n}{2} - n$ consultas de adyacencia.

Problema 4. (★★★) Demuestre que determinar si un grafo no dirigido de n nodos es conexo se requieren $\binom{n}{2}$ consultas de adyacencia.