Solución

Algoritmos y Estructuras de Datos III - 2C2024

14 de noviembre de 2024

Faltan los dibujos y algunas de las cosas que charlamos en clase, pero acá están las ideas generales para cada inciso

Ejercicio 1

a) Modelamos las computadoras y los cables como los vértices y las aristas de un grafo, respectivamente. Definimos $n = |S|$ y $m = |C|$. El ancho de banda entre dos computadoras es exactamente el mínimo ancho de banda de un cable en un camino maximin entre esas dos computadoras. Se calcula entonces un árbol maximin con un algoritmo de AGMáx de complejidad $O(m \log n)$ y se lo enraiza en s_c. Queremos encontrar la arista de mínimo ancho de banda entre las que conecten un amigo con la central. Para saber cuáles conectan con la central, calculamos recursivamente si hay algún amigo debajo de cada arista del árbol, por ejemplo, como hicimos en el ejercicio 2 b). Esto tiene complejidad $O(m)$. Luego, recorremos todas las aristas que conectan un amigo con la central para encontrar la de mínimo ancho de banda en $O(m)$, y retornamos ese valor. En total, la complejidad temporal es $O(m + m \log n) = O(m \log n)$.

b) Supongamos que $T_1 \cup T_2 \cup \{f\} = T_f$ no es AGM de $G \setminus e$, y entonces vale que existe un árbol generador T' de $G \setminus e$ tal que $c(T') < c(T_f)$. Llamemos T'_i al bosque contenido en T' que contiene los mismos vértices de T_i para $i \in \{1, 2\}$. A priori, no sabemos si T'_i es un árbol, solo que es un bosque.

Notemos que al agregar e a T' estamos conectando un vértice de T'_1 con uno de T'_2. Por lo tanto, al agregar e a T', se forma un ciclo que contiene vértices de T'_1 y de T'_2. En particular, este ciclo debe contener por lo menos una arista f' diferente de e que conecte un vértice de T'_1 con uno de T'_2. Como T'_1 y T'_2 tienen respectivamente los mismos vértices que T_1 y T_2, la arista f' también conecta un vértice de T_1 y uno de T_2 en G. En consecuencia, tendrá costo mayor o igual a f, por definición de f.

Definimos entonces $T'' = T' \cup \{e\} \setminus \{f'\}$, que es un grafo generador de G. Además, T'' tiene $n - 1$ aristas. Faltaría luego ver que T'' es conexo para poder decir que es un árbol generador de G. Para eso, tomemos dos vértices v y w de T', y veamos que están conectados por un camino en T''.

Si el camino simple que conecta v y w en T' no contiene a f' , entonces sigue existiendo en T''. Si, en cambio, el camino sí contiene a f', entonces v alcanza a uno de los extremos v' de f' sin pasar por f' y w alcanza el otro extremo w' de f' sin pasar por f'. Estos dos vértices que son extremos de f' pertenecen al ciclo que se forma al agregar e a T', por definición de f'. Por lo tanto, existe un camino simple entre v y w en T'' que primero alcanza v', luego alcanza w' pasando por la parte del ciclo que no contiene a f', y finalmente alcanza w. En conclusión, T'' es conexo, y por lo tanto es un árbol generador de G.

Ahora que tenemos un árbol generador T'' de G queremos probar que $c(T'') < c(T)$ para llegar a un absurdo. Sabemos que

$$\begin{aligned}
c(T'') &= c(T' \cup \{e\} \setminus \{f'\}) \\
&= c(T') + c(e) - c(f') \\
&\leq c(T') + c(e) - c(f) && (c(f') \geq c(f)) \\
&< c(T_f) + c(e) - c(f) && (c(T') < c(T_f)) \\
&= c(T).
\end{aligned}$$

Esto implica que T no es un AGM, lo cual es absurdo, y vino de suponer que T_f no es AGM de $G \setminus e$.

c) Si los sapos rompen una arista que no está en el árbol maximin, entonces los caminos maximin quedan iguales. Sino, por el inciso b) sabemos que el AGMáx del nuevo grafo se arma buscando la arista más pesada entre el subárbol desconectado y el resto. Removemos entonces una arista del AGMáx, y luego marcamos los nodos de alguna de las componentes conexas resultantes T_1 en el árbol usando algún algoritmo de búsqueda, por ejemplo, BFS. Esto tiene complejidad temporal $O(n + m)$. Después, recorremos todas las aristas para encontrar la más pesada entre las que conectan un nodo de T_1 con otro que no está en T_1. Esto toma $O(m)$. Una vez encontrada, se agrega al árbol, y se calculan los anchos de banda de los amigos como se hizo en el inciso a) en $O(n)$ (porque ya tenemos el árbol armado). Como hay que hacer esto para cada eje del árbol, se requieren $O(nm)$ operaciones.