16) *Algoritmo de Prim.*

Entrada. Un grafo conexo ponderado.

Idea. Mantener un subgrafo conexo H e ir agregando nuevos vértices incidentes en aristas de peso mínimo.

Inicio. Elegir $v \in V(G)$. $V(H) = \{v\}$, $E(H) = \varnothing$.

Iteración. Mientras $V(H) \neq V(G)$, elegir la arista e de menor peso entre el conjunto de aristas que tienen un extremo en $V(H)$ y otro en $V(G)\backslash V(H)$. Si la arista elegida es $e = uw$ con $u \in V(H)$ y $w \in V(G)\backslash V(H)$, actualizamos $V(H) \leftarrow V(H) \cup \{w\}$, $E(H) \leftarrow E(H) \cup \{e\}$.

a) Probar que si G es un grafo conexo ponderado, el algoritmo de Prim produce un árbol recubridor de G de peso mínimo.

b) Dar un árbol recubridor de peso mínimo para cada uno de los grafos del ejercicio 15, utilizando el algoritmo de Prim.

a) Sea G un grafo conexo ponderado y sea H el subgrafo que genera el algoritmo. Primero observemos que H es un árbol recubridor de G. En efecto, en cada iteración agregamos una arista incidente en un nuevo vértice que añadimos a H manteniendo H conexo y acíclico, por lo que H es un árbol. Además, como el algoritmo finaliza cuando $V(G) = V(H)$, H resulta un árbol recubridor de G. Veamos ahora que H es de mínimo peso.

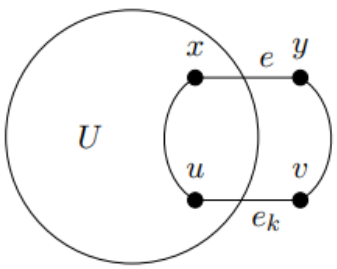
Sean $e_1, \ldots, e_{n-1}$ las aristas de H ordenadas en el mismo orden en que fueron agregadas. Para un árbol recubridor T cualquiera de G, definimos $a(T) = \max_{i\in[n]}\{i : e_j \in T \text{ para todo } j < i\}$. Si $T = H$, entonces $a(T) = n$. Entre todos los árboles recubridores de G de peso mínimo, elegimos H' de manera que $a(H') = \max\{a(T) : T \text{ árbol recubridor de } G\}$.

Sea $k = a(H')$. Tenemos que $e_i \in E(H') \cap E(H)$ para todo $i < k$.

Supongamos que H no es un árbol recubridor de peso mínimo. Entonces, $H \neq H'$, $k \leqslant n-1$, y $e_k \in E(H)\backslash E(H')$.

Sea U el conjunto de vértices que ya han sido agregados a H antes de agregar la arista e_k. Por cómo son elegidas las aristas en cada iteración del algoritmo de Prim, $e_k = uv$ para ciertos vértices u y v con $u \in U$ y $v \notin U$.

Por otro lado, como H' es un árbol recubridor de G, para estos vértices u y v existe un único u,v-camino simple P en H'. Sea $e = xy$ la primer arista de este u,v-camino que tiene un extremo, x, en U y otro, y, fuera de U.



Observemos que si agregamos la arista e_k a H', se genera un ciclo, concatenando P con esta arista. La arista e pertenece a este ciclo. Luego, no es una arista de corte en $H' \cup e_k$.

Borrando la arista e, obtenemos $T = (H' \cup e_k) \backslash e$ conexo y acíclico (pues rompemos el único ciclo que se formó al agregar e_k) y resulta un árbol recubridor.

Notemos que $w(T) = w(H') + w(e_k) - w(e)$.

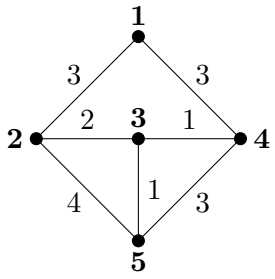
Como en el paso k del algoritmo de Prim elegimos la arista e_k antes que la arista e, resulta que $w(e_k) \leqslant w(e)$. Por otro lado, como T es un árbol recubridor y H' es un árbol recubridor de peso mínimo, resulta que $w(H') \leqslant w(T)$. Luego,

$$w(H') \leqslant w(T) = w(H') + w(e_k) - w(e) \leqslant w(H') \quad \text{pues } w(e_k) - w(e) \leqslant 0$$

Entonces, $w(T) = w(H')$ y T es también un árbol recubridor de peso mínimo. Pero $e_i \in E(T) \cap E(H)$ para todo $i \in [k]$. Luego, $a(T) \geqslant k + 1$. Esto contradice la elección de H'.

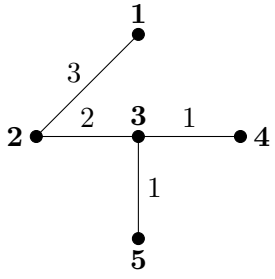
La contradicción surge de suponer que H no era de peso mínimo. Por lo tanto, concluimos que el algoritmo de Prim devuelve un árbol recubridor de peso mínimo de G.

b) Consideremos el grafo ponderado G de la figura.



Utilicemos el algoritmo de Prim para obtener un árbol recubridor de peso mínimo de G.

Como vértice inicial elegimos el vértice 1. Luego, agregamos al subgrafo H las aristas en el siguiente orden: $\{1,2\}, \{2,3\}, \{3,4\}, \{3,5\}$. Obteniendo el siguiente árbol de peso 7.



1) $V(H) = \{1\}$, $E(H) = \emptyset$.
2) $V(H) = \{1,2\}$, $E(H) = \{12\}$
3) $V(H) = \{1,2,3\}$, $E(H) = \{12,23\}$.
4) $V(H) = \{1,2,3,4\}$, $E(H) = \{12,23,34\}$.
5) $V(H) = V(G)$, $E(H) = \{12,23,34,35\}$.
Finalizo porque $V(H) = V(G)$.

ESCUELA DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICA
Av. Pellegrini 250. Rosario
+54 0341 - 480 2649 internos 216 - 119

17) Determinar si cada una de las siguiente afirmaciones es verdadera o falsa, justificando adecuadamente.

a) Si todos los pesos en un grafo conexo ponderado G son diferentes, entonces G admite un único árbol recubridor de peso mínimo.

b) Si todos los pesos en un grafo conexo ponderado G son diferentes, entonces los árboles recubridores de G distintos tienen pesos distintos.

a) *Verdadero.*

Sea G un grafo conexo ponderado tal que todos los pesos de sus aristas son diferentes. Para cada arista $e \in E(G)$, denotemos por $w(e)$ al peso de la arista e, y sea $w(G) = \sum_{e \in E(G)} w(e)$.

Como G es conexo, admite un árbol recubridor. Supongamos que G admite dos árboles recubridores de peso mínimo distintos T y T'.

Sea e la arista de menor peso en $E(T) \triangle E(T')$.

Supongamos que $e \in E(T) \backslash E(T')$. Por lo hecho en el ejercicio 12 (b), existe $e' \in E(T') \backslash E(T)$ tal que $(T' \cup e) \backslash e'$ es un árbol recubridor de G. Como $e' \in E(T) \triangle E(T')$ y todos los pesos de G son diferentes, $w(e) < w(e')$.

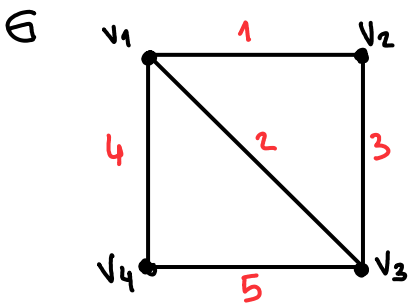
Pero notemos que

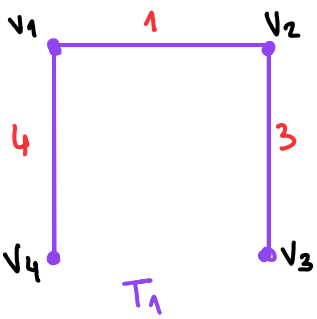
$$w((T' \cup e) \backslash e') = w(T') + w(e) - w(e') < w(T') \quad \text{pues } w(e) - w(e') < 0$$

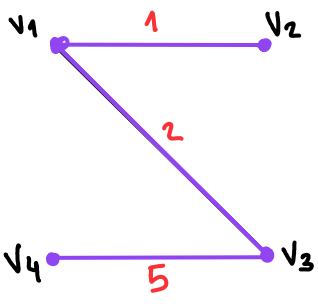
Así obtenemos que $w((T' \cup e) \backslash e') < w(T')$. Es decir, $(T' \cup e) \backslash e'$ es un árbol recubridor de menor costo que T'. Esto no es posible ya que T' es un árbol recubridor de peso mínimo.

Por lo tanto, concluimos que G tiene un único árbol recubridor de peso mínimo.

b) **Falso**







$c(T_1) = 1+3+4 = 8$

$c(T_2) = 1+2+5 = 8$

T1 y T2 son dos árboles recubridores de G distintos que tienen el mismo peso.

Algoritmo de Dijkstra

Entrada: Un grafo G ponderado con pesos no negativos y un vértice de inicio u. El peso de una arista xy es $\omega(xy)$ y si $xy \notin E(G)$, consideramos $\omega(xy) = \infty$.

Idea: Considerar un conj. S de vértices para los cuales hallamos un camino mínimo desde u, agrandando S hasta incluir todos los vértices. Tendremos una distancia arbitraria $t(z)$ desde u a cada $z \notin S$, hasta que la distancia mínima sea hallada.

Inicio: $S = \{u\}$, $t(u) = 0$, $t(z) = \omega(uz)\ \forall z \neq u$.

Iteración: Considerar $v \notin S$ con $t(v) = \min\{t(z) : z \notin S\}$.
Agregar v a S.
Explorar las aristas desde v y actualizar las etiquetas $t(z)$ para z vecino de v y $z \notin S$ con $t(z) = \min\{t(z), t(v) + \omega(vz)\}$.
Continuar la iteración hasta que $S = V(G)$ o hasta que $T(z) = \infty\ \forall z \notin S$.

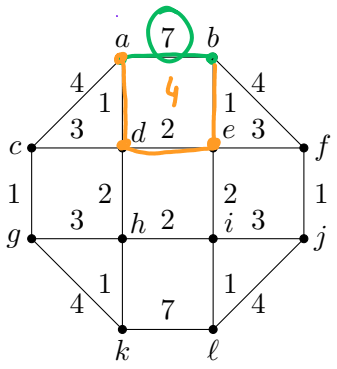
20) *a*) Aplicar el algoritmo de Dijkstra al grafo ponderado de la figura y determinar la distancia del vértice a a cada uno de los otros vértices.

$t(c) = \min\{t(c), t(d) + \omega(cd)\} =$
$= \min\{4, 1+3\} = 4$

$t(b) = \min\{t(b), t(d) + \infty\} = t(b)$

$t(e) = \min\{t(e), t(d) + \omega(de)\} =$
$= \min\{\infty, 1+2\} = 3$

$S = \{a, d\}$.
$t(a) = 0$
$t(b) = 7$
$t(c) = 4$
$t(d) = 1$
$t(v) = \infty\ \forall v \neq \dots$



b) Dar un camino de longitud mínima desde el vértice a a los vértices f, g y ℓ.

a) Inicializamos el algoritmo con el vértice a

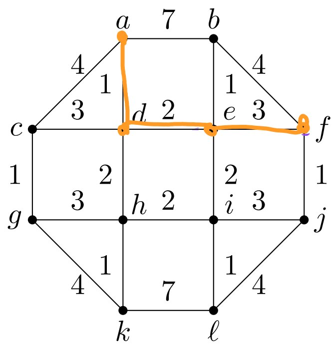
It.	a	b	c	d	e	f	g	h	i	j	k	ℓ
0	$(\mathbf{0}, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$
1	-	$(7, a)$	$(4, a)$	$(\mathbf{1}, \boldsymbol{a})$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$
2	-	$(7, a)$	$(4, a)$	-	$(\mathbf{3}, \boldsymbol{d})$	$(\infty, -)$	$(\infty, -)$	$(3, d)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$
3	-	$(4, e)$	$(4, a)$	-	-	$(6, e)$	$(\infty, -)$	$(\mathbf{3}, \boldsymbol{d})$	$(5, e)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$
4	-	$(\mathbf{4}, \boldsymbol{e})$	$(4, a)$	-	-	$(6, e)$	$(6, h)$	-	$(5, e)$	$(\infty, -)$	$(4, h)$	$(\infty, -)$
5	-	-	$(\mathbf{4}, \boldsymbol{a})$	-	-	$(6, e)$	$(6, h)$	-	$(5, e)$	$(\infty, -)$	$(4, h)$	$(\infty, -)$
6	-	-	-	-	-	$(6, e)$	$(5, c)$	-	$(5, e)$	$(\infty, -)$	$(\mathbf{4}, \boldsymbol{h})$	$(\infty, -)$
7	-	-	-	-	-	$(6, e)$	$(\mathbf{5}, \boldsymbol{c})$	-	$(5, e)$	$(\infty, -)$	-	$(11, k)$
8	-	-	-	-	-	$(6, e)$	-	-	$(\mathbf{5}, \boldsymbol{e})$	$(\infty, -)$	-	$(11, k)$
9	-	-	-	-	-	$(\mathbf{6}, \boldsymbol{e})$	-	-	-	$(8, i)$	-	$(6, i)$
10	-	-	-	-	-	-	-	-	-	$(7, f)$	-	$(\mathbf{6}, \boldsymbol{i})$
11	-	-	-	-	-	-	-	-	-	$(\mathbf{7}, \boldsymbol{f})$	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-

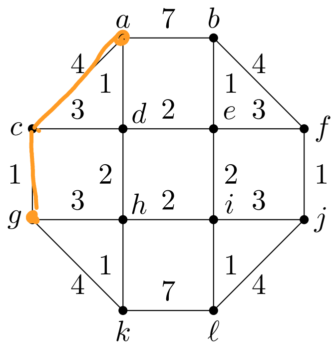
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICA
Av. Pellegrini 250. Rosario
+54 0341 - 480 2649 internos 216 - 119

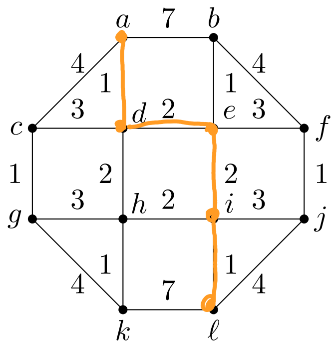
Luego, la longitud de un camino más corto de a a cada vértice es:

$$\begin{array}{lll} d(a,a)=0 & d(a,b)=4 & d(a,c)=4 \\ d(a,d)=1 & d(a,e)=3 & d(a,f)=6 \\ d(a,g)=5 & d(a,h)=3 & d(a,i)=5 \\ d(a,j)=7 & d(a,k)=4 & d(a,l)=6 \end{array}$$

b) A partir de la ejecución, podemos determinar caminos de longitud mínima del vértice a a cada uno de los demás vértices, observando las etiquetas, y reconstruyendo hacia atrás dichos caminos.

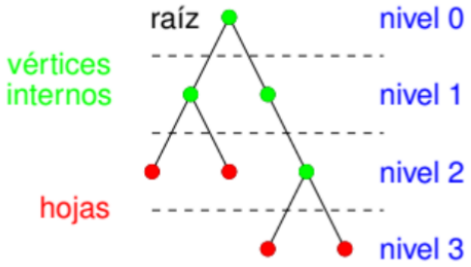






Árboles binarios

Definición 1. *Un árbol* ***enreaizado*** T *es un árbol con un único vértice distinguido* r*, llamado* ***raíz*** *de* T*. Un árbol enraizado es* ***binario*** *si todo vértice tiene a lo sumo dos hijos. Si en particular todo vértice tiene grado 0 o 2 hijos el árbol es* ***binario completo***.



24) Un árbol enraizado es m-*ario* (con $m \in \mathbb{N}$) si todo vértice tiene a lo sumo m hijos. Si en particular todo vértice tiene 0 o m hijos, se dice m-*ario completo*. Sea T un árbol m-ario completo con i vértices internos.

   a) Determinar la cantidad de hojas de T.

   b) Determinar la cantidad de vértices de T.

Sea T un árbol $m-$ario completo con i vértices internos y l hojas.

a) Como T es un árbol $m-$ario completo, cada uno de los i vértices internos tiene exactamente m hijos. Luego, hay im vértices que son hijos.

Por otro lado, la cantidad de vértices que son hijos son todos salvo la raíz. Esto es, $l + i - 1$. Luego, tenemos que $im = l + i - 1$, y como consecuencia $l = i(m-1) + 1$.

$\therefore$ T tiene $i(m-1)+1$ hojas.

b) La cantidad de vértices es $|V(T)| = i + l = im + 1$.

ESCUELA DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE MATEMÁTICA
Av. Pellegrini 250. Rosario
+54 0341 - 480 2649 internos 216 - 119