El centro de un árbol

Radio, centro y diámetro pelando capas de hojas

Carlos A Delgado S

Pontificia Universidad Javeriana Cali

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

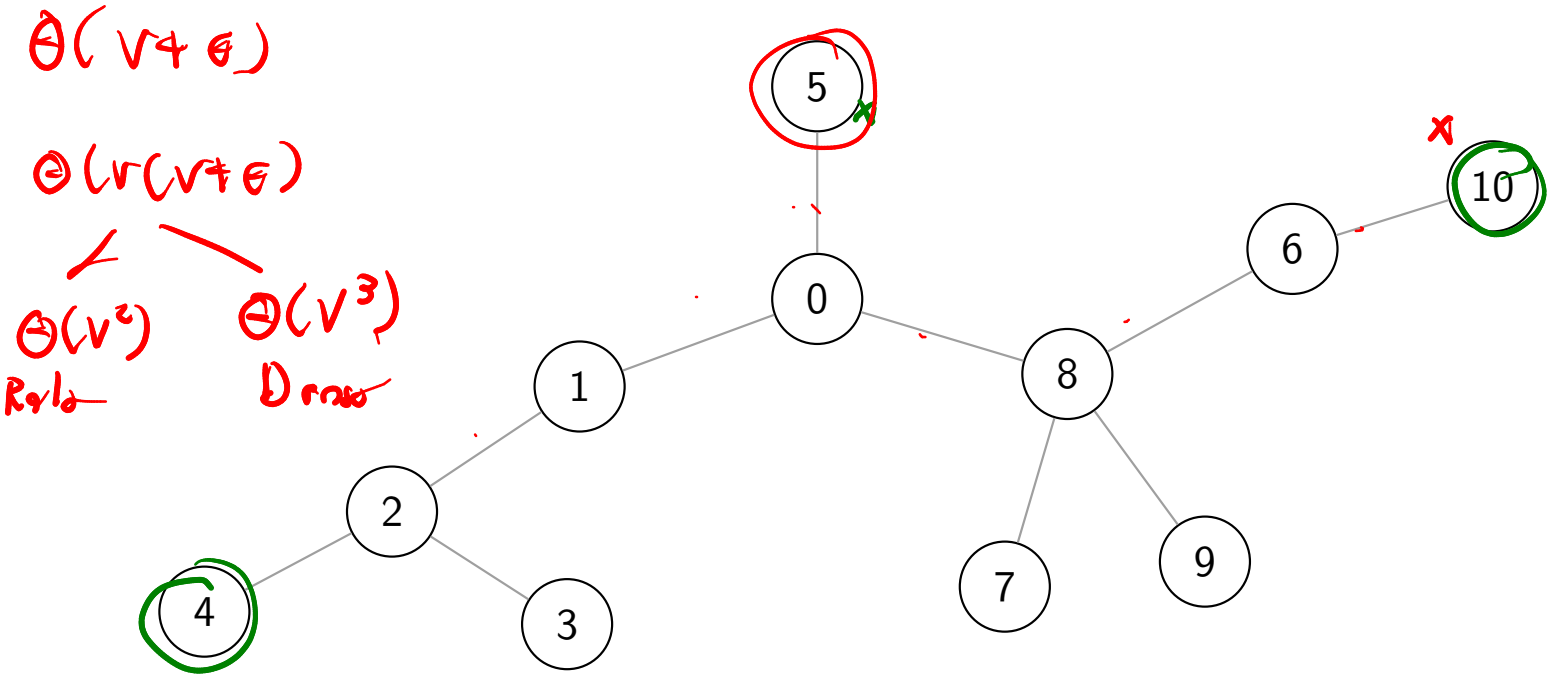
7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

El árbol que veníamos usando



Antes de calcular nada

La excentricidad $\varepsilon(v)$ es la distancia al vértice más lejano, el radio es la menor de todas y los centros son los vértices que la alcanzan. Mírelo y apueste por uno: ¿cuál queda más al medio?

Dos preguntas que quedaron abiertas

La primera

¿Por qué el vértice más lejano a uno cualquiera es siempre un extremo de un camino más largo del árbol?

La segunda

Si se quitan todas las hojas de un árbol y se repite, ¿qué queda al final y cuántas vueltas toma?

La primera, en una línea

Por qué funcionan las dos pasadas

Sea u el más lejano a un v cualquiera y sea $x-y$ un camino más largo. Si u no fuera extremo de ninguno, el camino de v a u se podría reemplazar por uno hacia x o hacia y sin acortarse, y quedaría un camino más largo que $x-y$. Eso contradice que $x-y$ sea de los más largos.

En el árbol de once vértices

Desde el 5, el más lejano es el 4, a cinco aristas. Y el 4 es extremo de un camino más largo: el que va del 4 al 10, que mide seis. La primera pasada cae en una punta del diámetro aunque haya empezado en cualquier parte.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

La forma cara de contestar

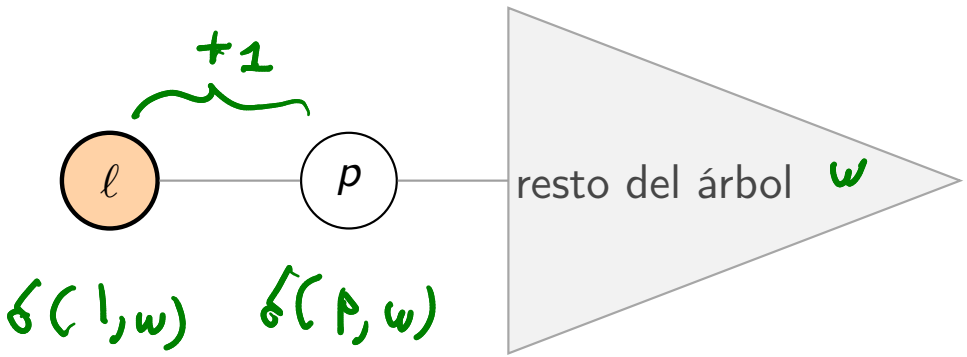
Una amplitud desde cada vértice

Con n búsquedas se tienen las n excentricidades, y de ahí salen el radio, el diámetro y el centro. En un árbol cada búsqueda cuesta $\Theta(n)$, así que el total es $\Theta(n^2)$.

Con los números de este árbol

Once vértices, once búsquedas, 121 pasos para enterarse de que el centro es un solo vértice.

¿Hace falta conocerlas todas?



La cuenta

Sea ℓ una hoja y p su único vecino. Todo camino que sale de ℓ pasa por p, así que para todo $w \neq \ell$ se tiene $\delta(\ell, w) = 1 + \delta(p, w)$. Tomando el máximo,

$$\varepsilon(\ell) = 1 + \varepsilon(p).$$

La consecuencia

Con tres vértices o más, una hoja siempre tiene excentricidad mayor que su vecino. Ninguna hoja es centro.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Teorema de Jordan, 1869

Teorema (Centro de un árbol)

Sea $T = (V, E)$ *un árbol y sea* $L \subseteq V$ *su conjunto de hojas.*

- *Si* $|V| \leq 2$, *entonces* $\mathcal{C}(T) = V$.
- *Si* $|V| > 2$, *entonces* $\mathcal{C}(T) = \mathcal{C}(T \setminus L)$.

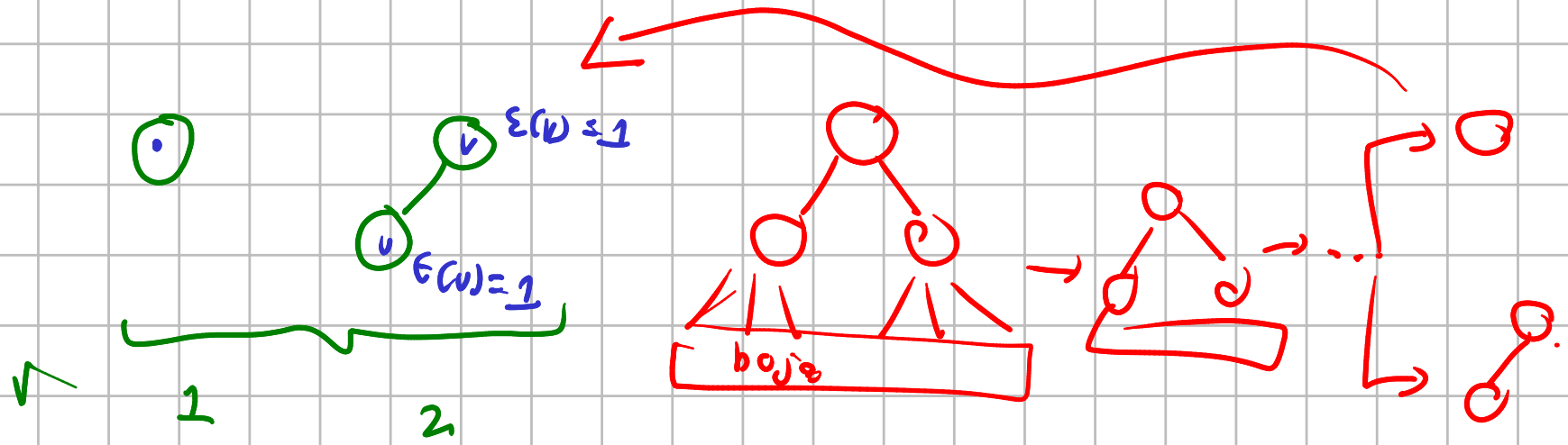
Lo que dice

Quitar todas las hojas a la vez no mueve el centro. Y un árbol de uno o dos vértices es su propio centro.

Sea $T = (V, E)$ un árbol y sea $L \subseteq V$ su conjunto de hojas.

- *Si $|V| \leq 2$, entonces $\mathcal{C}(T) = V$.*
- *Si $|V| > 2$, entonces $\mathcal{C}(T) = \mathcal{C}(T \setminus L)$.*

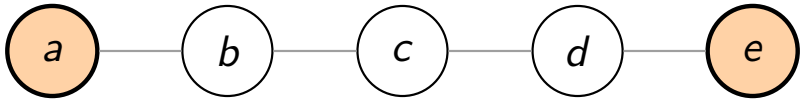
hoja



Dónde termina el camino más largo desde un interno

Siempre en una hoja

Para cualquier v que no sea hoja, el camino que realiza $\varepsilon(v)$ termina en una hoja: si terminara en un vértice interno, se podría seguir un paso más y el camino sería más largo.



En el camino de cinco

$\varepsilon(c) = 2$, y el camino que lo realiza termina en a o en e, que son las dos hojas. Desde b pasa lo mismo, y termina en e.

Todas bajan en uno

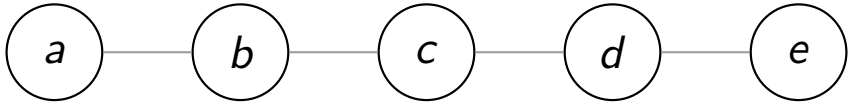
Qué le pasa a cada excentricidad

Al quitar todas las hojas, ese camino pierde exactamente su último paso, de modo que

$$\varepsilon_{T\setminus L}(v) = \varepsilon_T(v) - 1$$

para todo vértice interno v. Si todas bajan en uno, el mínimo baja en uno, pero los vértices donde se alcanza son los mismos. El radio cambia; el centro no.

Comprobado en el camino de cinco



	a	b	c	d	e	radio	centro
árbol completo	4	3	**2**	3	4	2	$\{c\}$
sin $\{a, e\}$		2	**1**	2		1	$\{c\}$
sin $\{b, d\}$			**0**			0	$\{c\}$

Lo que muestra la tabla

Cada fila baja en uno respecto de la anterior, en todas las columnas que siguen vivas, y la columna donde está el mínimo nunca cambia.

Del teorema al algoritmo

Se aplica una y otra vez

El teorema deja el mismo problema sobre un árbol más pequeño, así que se vuelve a aplicar. Cada vuelta quita por lo menos dos vértices, de modo que el proceso termina, y termina con uno o dos vértices, que son el centro.

En el camino de cinco, dos vueltas

Primero se van a y e, después b y d, y queda c. El centro de los tres árboles que aparecieron es el mismo, $\{c\}$, y lo único que cambió fue el radio: 2, 1 y 0.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

El algoritmo

En palabras

Centro(T):

1. $L \leftarrow$ las hojas de T
2. **mientras** $|V(T)| > 2$:

2.1 $\quad V(T) \leftarrow V(T) \setminus L$

2.2 $\quad L \leftarrow$ las hojas de lo que quedó

3. **devolver** $V(T)$

Lo que hay que resolver para programarlo

Quién es hoja cambia en cada vuelta, y borrar vértices de verdad costaría más que el propio algoritmo.

No se borra nada: se baja el grado

La idea

Se guarda el grado de cada vértice. Quitar una hoja es restarle uno al grado de su vecino. El que queda en grado uno pasa a ser hoja de la capa siguiente, y el que ya se fue se marca con grado cero para no volver a tocarlo.

Es el truco del orden topológico

Allá era el grado de entrada y la condición era llegar a cero.

Los grados y la primera capa

```
9  def capaDeHojas(arbol, grado):
10     # Los vertices vivos que tienen un solo vecino vivo.
11     capa = []
12     for v in arbol:
13         if grado[v] <= 1:
14             capa.append(v)
15     return capa
```

Una sola pasada

El grado sale del largo de la lista de adyacencia, y la primera capa de los que tienen grado uno. En un árbol de un solo vértice el grado es cero, y por eso la condición es ≤ 1.

Quitar una capa

```
18 def pelarCapa(arbol, grado, capa):
19     # Quita la capa entera y devuelve las hojas que quedan al hacerlo.
20     siguiente = []
21     for v in capa:
22         for u in arbol[v]:
23             if grado[u] > 0:
24                 grado[u] = grado[u] - 1
25                 if grado[u] == 1:
26                     siguiente.append(u)
27         # el vertice no se borra: se marca con grado cero
28         grado[v] = 0
29     return siguiente
```

El bucle

```
32 def centroArbol(arbol):
33     # Devuelve el centro y cuantas capas se quitaron para llegar a el.
34     grado = {}
35     for v in arbol:
36         grado[v] = len(arbol[v])
37     capa = capaDeHojas(arbol, grado)
38     vivos = len(arbol)
39     vueltas = 0
40     while vivos > 2:
41         vivos = vivos - len(capa)
42         capa = pelarCapa(arbol, grado, capa)
43         vueltas = vueltas + 1
44     return capa, vueltas
```

Qué sostiene cada vuelta

Al entrar al ciclo, capa son las hojas de lo que sigue vivo y `vivos` es cuántos quedan. Cada vuelta quita la capa entera y deja lista la siguiente, así que al salir capa es el centro.

La capa completa, antes de mirar la siguiente

El error que rompe el algoritmo

Si se mete cada vértice nuevo a la misma lista que se está recorriendo, un vértice interno puede pelarse en la misma vuelta en que se pelaron sus hijos. El árbol se consume de un lado y el centro sale corrido.

Cómo lo evita el código

Los que llegan a grado uno entran a `siguiente`, que es otra lista. Solo al terminar la vuelta pasa a ser la capa actual.

Lo que cuesta

Lineal

Cada vértice entra a una capa a lo sumo una vez y cada arista se mira dos veces, una por cada punta. En total $\Theta(n)$.

vértices	una amplitud por vértice	pelando hojas
11	121	11
1 000	1 000 000	1 000
100 000	10^{10}	100 000

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

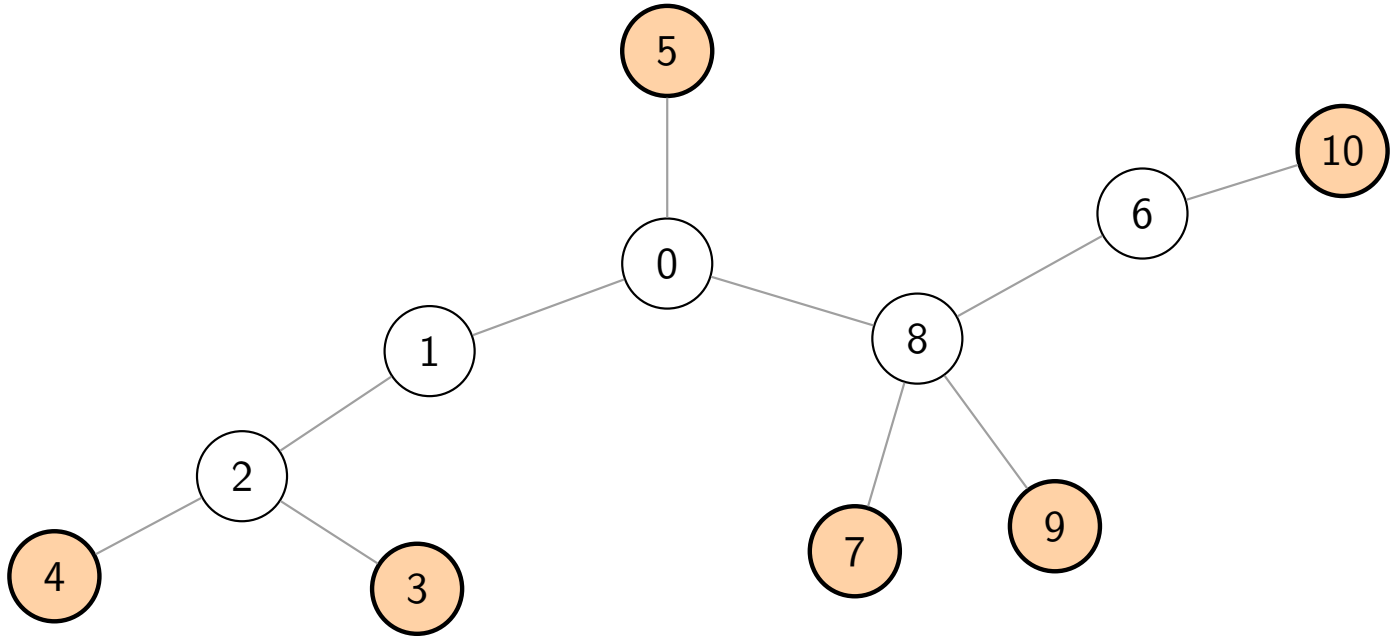
7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

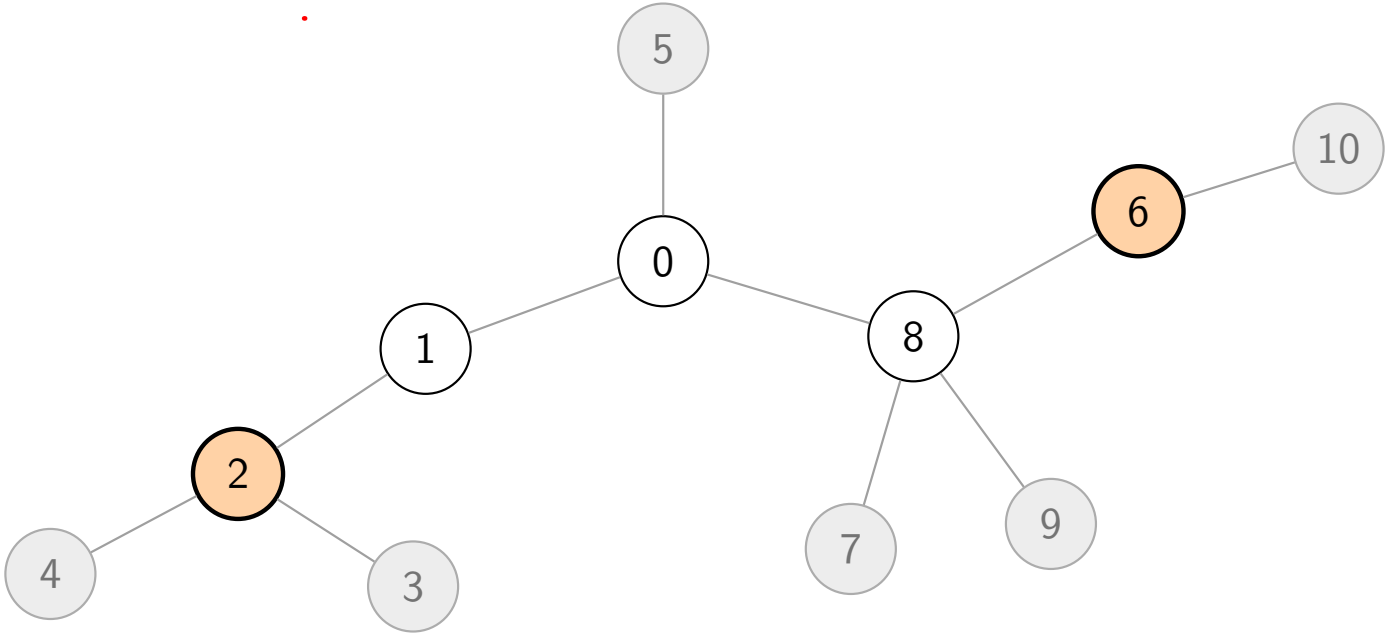
Primera capa: los seis de grado uno



Quiénes

3, 4, 5, 7, 9 y 10. Quedan once menos seis, o sea cinco vértices, que todavía son más de dos.

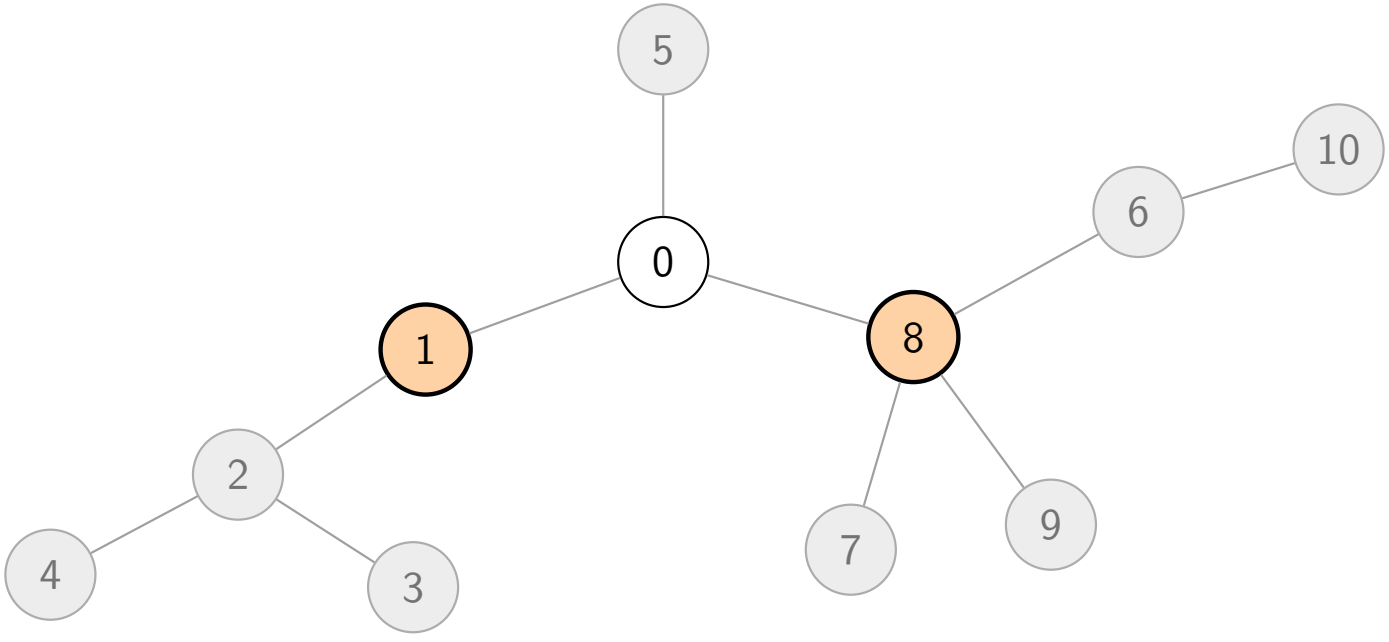
Segunda capa



Qué pasó con los grados

El 2 perdió a 3 y a 4, y pasó de grado tres a grado uno. El 6 perdió a 10 y quedó en uno. El 8 perdió a 7 y a 9: bajó de cuatro a dos, todavía no es hoja.

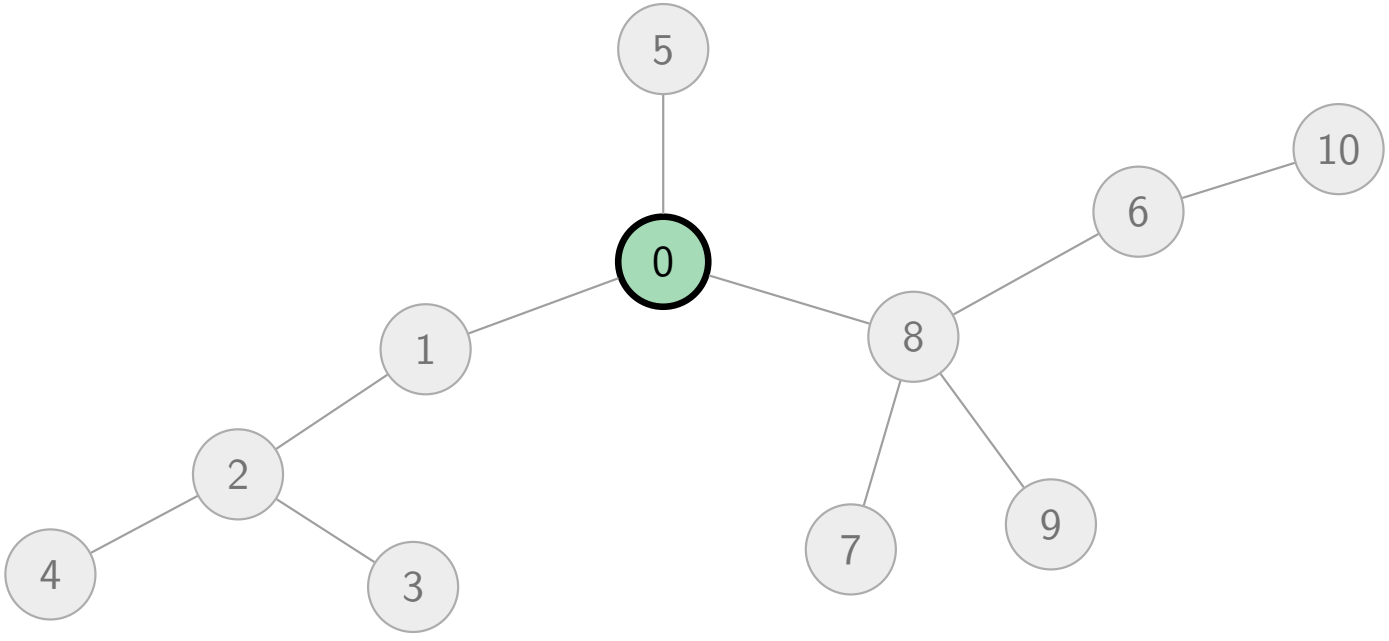
Tercera capa



Qué pasó

Al irse el 2, el 1 queda con un solo vecino. Al irse el 6, el 8 también.
Quedan tres vértices vivos, 0, 1 y 8, y tres es más de dos: hay otra vuelta.

Lo que queda



El centro

Queda el vértice 0, y es el mismo que da la cuenta de las excentricidades.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

¿Cuántas vueltas dio?

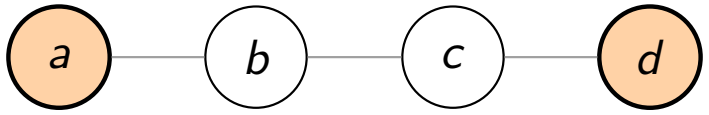
La pregunta

El algoritmo dio tres vueltas sobre este árbol, y su radio es tres. ¿Es casualidad, o el número de vueltas es siempre el radio?

Antes de contestar

Pruébelo en el camino de cuatro vértices. ¿Cuántas vueltas da? ¿Cuál es su radio?

El camino de cuatro dice que no



Una sola vuelta

Se van a y d, quedan b y c, que son dos, y el algoritmo para. Una vuelta.

Pero el radio es dos

$\varepsilon(a) = \varepsilon(d) = 3$ y $\varepsilon(b) = \varepsilon(c) = 2$. El radio es 2 y el centro es $\{b, c\}$.

La regla completa

Lo que cuentan las vueltas

Cada vuelta acerca el borde en un paso, así que las vueltas son $\lfloor diam/2 \rfloor$. Lo que falta lo dice cuántos vértices quedaron.



	un centro	dos centros
radio	vueltas	vueltas + 1
diámetro	2· vueltas	2· vueltas + 1

Por qué dos casos

Con diámetro par la mitad del camino más largo cae en un vértice, y con diámetro impar cae entre dos.

Las tres medidas de una sola pasada

```
47 def radioDiametro(arbol):
48     # El radio y el diametro salen de las vueltas y de cuantos centros
       ↪  quedan.
49     (centro, vueltas) = centroArbol(arbol)
50     if len(centro) == 1:
51         radio = vueltas
52         diametro = 2 * vueltas
53     else:
54         radio = vueltas + 1
55         diametro = 2 * vueltas + 1
56     return radio, diametro, centro
```

Arbol

$E \longrightarrow V$

$G(V, E)$

Qué se gana

El centro, el radio y el diámetro de un árbol en $\Theta(n)$, sin calcular ninguna excentricidad y sin escoger raíz.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Tres formas de equivocarse

Usarlo en un grafo con ciclos

Un ciclo no tiene hojas: el algoritmo se queda quieto y devuelve lo que no es. El radio y el centro de un grafo cualquiera piden las distancias de todos contra todos.

Parar cuando queda uno

La condición es $|V| > 2$. Si se exige llegar a un solo vértice, los árboles de diámetro impar entran en un ciclo infinito o devuelven uno de los dos centros, que es la mitad de la respuesta.

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Ejercicio 1: en parejas

Construya el árbol que le pidan

1. Un árbol de diez vértices con dos centros.
2. Un árbol de diez vértices cuyo centro sea una sola hoja del árbol que queda tras la primera vuelta.
3. Un árbol donde el algoritmo dé exactamente cuatro vueltas.

Y después

Intercambien los árboles con la pareja de al lado y verifiquen el centro del otro pelando hojas.

Ejercicio 2: cambiar un vuelo

UVa 11695, Flight Planning

Hay n ciudades y $n-1$ vuelos que las conectan, de modo que entre dos cualesquiera hay una sola ruta. Se quita un vuelo y se pone otro, buscando que el mayor número de trasbordos entre dos ciudades quede lo más pequeño posible. Se pide ese número y los dos vuelos.

El caso del enunciado

Cuatro ciudades en fila, $1-2$, $2-3$, $3-4$. El diámetro es 3; quitando $3-4$ y volviendo a pegar el 4 al 2 queda en 2.

Dónde está

Enunciado y envío: `https://vjudge.net/problem/UVA-11695`

Cómo leer la entrada

```cpp
int casos;
scanf("%d", &casos);
for (int caso = 0; caso < casos; caso++) {
  int n;
  scanf("%d", &n);
  vector<int> a(n - 1), b(n - 1);
  for (int i = 0; i < n - 1; i++) {
    scanf("%d %d", &a[i], &b[i]);
  }
  /* aqui se prueban las n-1 aristas */
}
```

Dos detalles

Las ciudades vienen numeradas desde 1, así que o se les resta uno al leerlas o se reservan $n + 1$ casillas. La salida son tres líneas por caso: el diámetro nuevo, el vuelo que se quita y el que se pone.

Por dónde entrarle

Dos mitades y sus centros

Quitar una arista parte el árbol en dos. Volver a unirlos por sus centros es lo mejor que se puede hacer, y entonces el diámetro queda en el mayor entre d_1, d_2 y $\lceil d_1/2 \rceil + \lceil d_2/2 \rceil + 1$. Con $n \leq 2500$ alcanza con probar las $n - 1$ aristas y resolver cada mitad en tiempo lineal.

Para practicarlo antes

La actividad del tema prueba cada corte sobre tres redes y muestra los dos pedazos con su diámetro y su centro: `https://cardel.github.io/notasUniversidad/actividades/ayg-puj/centro-arbol/vuelos.html`

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Lo que sigue

Juntar conjuntos sin recorrer el grafo

Lo que viene usa los árboles para otra cosa: representar conjuntos que se van uniendo, con una estructura que responde si dos elementos están juntos en tiempo casi constante.

Para investigar

Si cada conjunto es un árbol y la unión cuelga una raíz de la otra, ¿qué tan alto puede quedar? ¿Y si al buscar la raíz se aprovecha para colgar de ella todo lo que quedó en el camino?

Contenido

1 Lo que quedó pendiente

2 Una hoja nunca es el centro

3 El teorema del centro

4 El algoritmo de pelar hojas

5 La traza

6 De las vueltas al radio y al diámetro

7 Errores comunes

8 Ejercicios

9 Lo que sigue

10 Referencias

Referencias


Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. *Introduction to Algorithms*, 3rd ed. MIT Press, 2009. Apéndice B.5, páginas 1173 a 1177.


Jordan, C. *Sur les assemblages de lignes*. Journal für die reine und angewandte Mathematik, 70, 1869, páginas 185 a 190.


Halim, S., Halim, F., Effendy, S. *Competitive Programming 4*. Lulu, 2020. Capítulo 4: propiedades de los árboles, diámetro y centro.


van Steen, M. *Graph Theory and Complex Networks*. Maarten van Steen, 2010. Capítulo 5: árboles, excentricidad y centro.


UVa Online Judge, problemas 10308 y 11695, por vjudge. `https://vjudge.net`