CI2613: Algoritmos y Estructuras III

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

Enero-Marzo 2015

Búsqueda en amplitud (BFS)

Búsqueda en amplitud (BFS)

Búsqueda en amplitud o *breadth-first search* explora el grafo $G = (V, E)$ a partir de un vértice "fuente" s de **forma sistemática** para descubrir todos los vértices que son **alcanzables** desde s

Búsqueda en amplitud calcula:

– la **distancia** (mínimo número de aristas) desde s hasta todos los vértices alcanzables desde s

– produce un árbol T (llamado un **árbol breadth-first**) que contiene a todos los vértices alcanzables desde s (incluyendo s)

Para todo vértice v en el árbol T, el **único camino** que conecta s con v en T es un **camino más corto** de s a v en el grafo $G = (V, E)$

Búsqueda en amplitud (BFS)

BFS trabaja en grafos dirigidos y no dirigidos

Su nombre se debe a que todos los vértices a distancia k (desde s) son descubiertos antes que cualquier vértice a distancia $k + 1$ (desde s)

BFS mantiene información de la búsqueda asignando **colores** (blanco, gris y negro) a los vértices

Búsqueda en amplitud (BFS)

Inicialmente, todos los vértices son blancos; luego pueden hacerse grises para después pasar a negros

Un vértice es **descubierto** la primera vez que se encuentra durante la búsqueda, en cuyo caso deja de ser blanco

Vértices grises y negros han sido descubiertos:

– Si u es negro y $(u,v) \in E$, entonces v es gris o negro; i.e. todos los vértices adyacentes a vértices negros no son blancos y por lo tanto han sido descubiertos

– Los vértices grises pueden tener vértices adyacentes blancos; ellos representan la frontera entre los vértices descubiertos y los no descubiertos

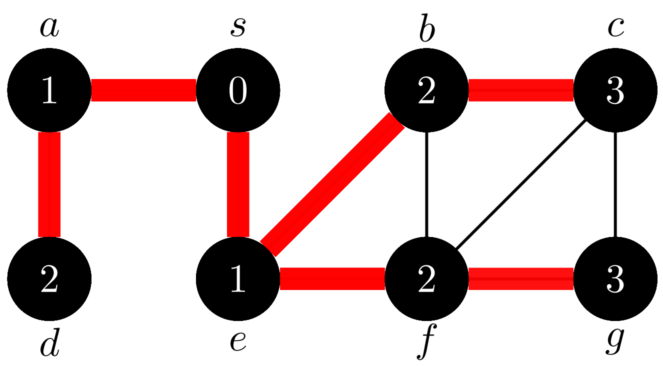
Búsqueda en amplitud: Pseudocódigo

```
breadth-first-search(Vertice s):
    % inicialización
    foreach Vertice u
        color[u] = Blanco
        d[u] = ∞               % distancia desde s a u
        π[u] = null            % padre de u en el árbol BF

    Queue q  % cola FIFO
    color[s] = Gris
    d[s] = 0
    q.enqueue(s)

    % búsqueda iterativa
    while !q.empty()
        Vertice u = q.dequeue()
        foreach Vertice v in adyacentes[u]
            if color[v] == Blanco
                color[v] = Gris
                d[v] = d[u] + 1
                π[v] = u
                q.enqueue(v)
        color[u] = Negro
```


Búsqueda en amplitud: Ejemplo



Q: —

Búsqueda en amplitud: Pseudocódigo

```
void breadth-first-search(Vertice s):
    % inicialización
    foreach Vertice u
        color[u] = Blanco
        d[u] = ∞               % distancia desde s a u
        π[u] = null            % padre de u en el árbol BF

    Queue q  % cola FIFO
    color[s] = Gris
    d[s] = 0
    q.enqueue(s)

    % búsqueda iterativa
    while !q.empty()
        Vertice u = q.dequeue()
        foreach Vertice v in adyacentes[u]
            if color[v] == Blanco
                color[v] = Gris
                d[v] = d[u] + 1
                π[v] = u
                q.enqueue(v)
        color[u] = Negro
```

Búsqueda en amplitud: Análisis de tiempo

Entrada: grafo $G = (V, E)$ representado con **listas de adyacencia**
(Tamaño de la entrada es $O(V + E)$)

Utilizamos la técnica de **análisis agregado**:

1. Después de inicialización, BFS no "pinta" ningún vértice de blanco
2. Por lo tanto, las líneas 17 y 18 implican que todo vértice es **encolado** (y por lo tanto **decolado**) a lo sumo 1 vez
3. Las operaciones de encolar/decolar toman tiempo $O(1)$; por lo tanto, el tiempo total (agregado) de estas operaciones es $O(V)$
4. Cada lista de adyacencia es recorrida a lo suma 1 vez (al decolar). Como el total de elementos en las listas es $O(E)$, el tiempo total invertido en recorrer las listas de adyacencia es $O(E)$
5. El tiempo para la inicialización es $O(V)$

El tiempo total de BFS es $O(V + E)$ (i.e. **tiempo lineal**)

Caminos más cortos

Considere un grafo $G = (V, E)$ (dirigido o no dirigido)

La **distancia de camino más corto** $\delta(s, v)$ de s a v es el mínimo número de aristas en un camino de s a v (si no existe dicho camino, definimos $\delta(s, v) = \infty$)

Un camino de s a v cuya longitud sea $\delta(s, v)$ es un **camino más corto**. (Nota: pueden existir múltiples caminos más cortos de s a v)

Caminos más cortos: Propiedades

Lema

Sea $G = (V, E)$ un grafo (dirigido o no dirigido), y sea s un vértice de G. Para toda arista $(u, v) \in E$, $\delta(s, v) \leq \delta(s, u) + 1$

Prueba:

Si u no es alcanzable desde s, $\delta(s, u) = \infty$ y la desigualdad es cierta

Si u es alcanzable desde s, también lo es v. En este caso, un camino más corto de s a v no puede ser más largo que la longitud del camino más corto de s a u concatenado con la arista (u, v); i.e. $\delta(s, v) \leq \delta(s, u) + 1$ $\square$

Búsqueda en amplitud: Pseudocódigo

```
1  breadth-first-search(Vertice s):
2      % inicialización
3      foreach Vertice u
4          color[u] = Blanco
5          d[u] = ∞              % distancia desde s a u
6          π[u] = null           % padre de u en el árbol BF
7
8      Queue q  % cola FIFO
9      color[s] = Gris
10     d[s] = 0
11     q.enqueue(s)
12
13     % búsqueda iterativa
14     while !q.empty()
15         Vertice u = q.dequeue()
16         foreach Vertice v in adyacentes[u]
17             if color[v] == Blanco
18                 color[v] = Gris
19                 d[v] = d[u] + 1
20                 π[v] = u
21                 q.enqueue(v)
22         color[u] = Negro
```


Búsqueda en amplitud: Correctitud

Veremos que al finalizar BFS, $\delta(s, v) = d[v]$ para todo $v \in V$

Lema

Sea $G = (V, E)$ un grafo (dirigido o no), y suponga que BFS se corre desde $s \in V$. Al terminar BFS, $d[v] \geq \delta(s, v)$ para todo $v \in V$

Prueba: Por **inducción en el número de encolamientos**

Tesis: $d[v] \geq \delta(s, v)$ para todo $v \in V$

Caso base: después del primer encolamiento, $d[s] = \delta(s, s) = 0$ y $d[v] = \infty \geq \delta(s, v)$ para $v \neq s$. La tesis se cumple

Búsqueda en amplitud: Correctitud

Veremos que al finalizar BFS, $\delta(s, v) = d[v]$ para todo $v \in V$

Lema

Sea $G = (V, E)$ un grafo (dirigido o no), y suponga que BFS se corre desde $s \in V$. Al terminar BFS, $d[v] \geq \delta(s, v)$ para todo $v \in V$

Prueba: Por **inducción en el número de encolamientos**

Tesis: $d[v] \geq \delta(s, v)$ para todo $v \in V$

Paso inductivo: considere el encolamiento de v (descubierto al recorrer la lista de adyancencia de u). Justo antes del encolamiento en la línea 21:

$$d[v] = d[u] + 1 \geq \delta(s, u) + 1 \geq \delta(s, v)$$

El vértice v es entonces encolado, y más nunca será encolado de nuevo porque se pinta de gris. Las líneas 18-21 sólo se ejecutan para vértices blancos, por lo tanto $d[v]$ no vuelve a cambiar $\square$

© 2014 Blai Bonet CI2613

Búsqueda en amplitud: Correctitud

La cola contiene vértices con a lo sumo dos valores distintos para d

Lema

Suponga que durante la ejecución de BFS sobre un grafo $G = (V, E)$, la cola contiene $\langle v_1, v_2, \ldots, v_r \rangle$ donde v_1 y v_r son el primero y último de la cola. Entonces, (i) $d[v_r] \leq d[v_1] + 1$ y (ii) $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \ldots, r - 1$

Prueba: Por **inducción en el número de encolamientos/decolamientos**

Tesis: la condición se verifica después de cada operación

Caso base: después del primer encolamiento, la cola sólo contiene a s y la tesis es válida trivialmente

© 2014 Blai Bonet CI2613

Búsqueda en amplitud: Correctitud

La cola contiene vértices con a lo sumo dos valores distintos para d

Lema

Suponga que durante la ejecución de BFS sobre un grafo $G = (V, E)$, la cola contiene $\langle v_1, v_2, \ldots, v_r \rangle$ donde v_1 y v_r son el primero y último de la cola. Entonces, (i) $d[v_r] \leq d[v_1] + 1$ y (ii) $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \ldots, r - 1$

Prueba: Por **inducción en el número de encolamientos/decolamientos**

Tesis: la condición se verifica después de cada operación

Paso inductivo: debemos verificar que la condición se cumple después de encolar o decolar vértices

© 2014 Blai Bonet CI2613

Búsqueda en amplitud: Correctitud

La cola contiene vértices con a lo sumo dos valores distintos para d

Lema

Suponga que durante la ejecución de BFS sobre un grafo $G = (V, E)$, la cola contiene $\langle v_1, v_2, \ldots, v_r\rangle$ donde v_1 y v_r son el primero y último de la cola. Entonces, (i) $d[v_r] \leq d[v_1] + 1$ y (ii) $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \ldots, r-1$

Prueba: Por **inducción en el número de encolamientos/decolamientos**

Tesis: la condición se verifica después de cada operación

❶ *Decolar:* considere la cola $\langle v_1, v_2, \ldots, v_r\rangle$ antes de decolar v_1, y la cola $\langle v_2, \ldots, v_r\rangle$ después de decolarlo.

Para (i), antes de decolar v_1, la condición se verifica (por hipótesis), luego $d[v_r] \leq d[v_1] + 1 \leq d[v_2] + 1$.

Para (ii), por hipótesis, $d[v_i] \leq d[v_{i+1}]$ para $i = 2, \ldots, r-1$

Búsqueda en amplitud: Correctitud

La cola contiene vértices con a lo sumo dos valores distintos para d

Lema

Suponga que durante la ejecución de BFS sobre un grafo $G = (V, E)$, la cola contiene $\langle v_1, v_2, \ldots, v_r\rangle$ donde v_1 y v_r son el primero y último de la cola. Entonces, (i) $d[v_r] \leq d[v_1] + 1$ y (ii) $d[v_i] \leq d[v_{i+1}]$ para $i = 1, \ldots, r-1$

Prueba: Por **inducción en el número de encolamientos/decolamientos**

Tesis: la condición se verifica después de cada operación

❷ *Encolar:* después de encolar v (línea 21), la cola queda $\langle v_1, v_2, \ldots, v_{r+1}\rangle$ donde $v_{r+1} = v$. En ese momento, v es adyacente a u que ha sido removido de la cola (línea 15). Antes de decolar u la cola es $\langle u, v_1, \ldots, v_j\rangle$ con $0 \leq j \leq r$. Si $j = 0$, v_1 fue descubierto explorando u y $d[u] < d[v_1]$. Si $j > 0$, por hipótesis, $d[u] \leq d[v_1]$

Para (i), $d[v_{r+1}] = d[v] = d[u] + 1 \leq d[v_1] + 1$

Para (ii), por hipótesis, $d[v_r] \leq d[u] + 1 = d[v] = d[v_{r+1}]$ □

Búsqueda en amplitud: Correctitud

Corolario

Suponga que los vértices v_i and v_j son encolados durante la ejecución de BFS, y que v_i es encolado antes que v_j (los vértices se encolan a lo sumo 1 vez). Entonces, $d[v_i] \leq d[v_j]$ al momento de encolar v_j

Prueba: sea $v_1, v_2, \ldots$ todos los vértices encolados (en orden). Para $i \geq 1$,

Si v_i y v_{i+1} aparecen alguna vez simultaneamente en la cola, el Lema implica $d[v_i] \leq d[v_{i+1}]$ al momento de encolar v_{i+1}

Si no aparecen simultaneamente en la cola, v_{i+1} fue descubierto al explorar v_i. Por lo tanto, $d[v_{i+1}] = d[v_i] + 1 > d[v_i]$ al momento de encolar v_{i+1} □

Búsqueda en amplitud: Pseudocódigo

```
1  breadth-first-search(Vertice s):
2      % inicialización
3      foreach Vertice u
4          color[u] = Blanco
5          d[u] = ∞              % distancia desde s a u
6          π[u] = null           % padre de u en el árbol BF
7
8      Queue q  % cola FIFO
9      color[s] = Gris
10     d[s] = 0
11     q.enqueue(s)
12
13     % búsqueda iterativa
14     while !q.empty()
15         Vertice u = q.dequeue()
16         foreach Vertice v in adyacentes[u]
17             if color[v] == Blanco
18                 color[v] = Gris
19                 d[v] = d[u] + 1
20                 π[v] = u
21                 q.enqueue(v)
22         color[u] = Negro
```

Búsqueda en amplitud: Correctitud

Teorema

Sea $G = (V, E)$ un grafo (dirigido o no), y suponga que BFS se ejecuta desde el vértice $s \in V$. Entonces, BFS descrubre cada vértice v que es alcanzable desde s, y al terminar:

(i) $d[v] = \delta(s, v)$ *para todo* $v \in V$

(ii) *para cada* $v \in V$ *con* $v \neq s$*, existe un camino más corto de* s *a* v*, que contiene la arista* $(\pi[v], v)$

Prueba: primero mostramos (i) por **contradicción**. Suponga que para algún vértice v, $d[v] \neq \delta(s, v)$. Sea v un tal vértice con valor $\delta(s, v)$ mínimo

Claramente, $v \neq s$ y $d[v] \geq \delta(s, v)$ (por Lema). Por lo tanto, $d[v] > \delta(s, v)$

Mas aún, v es alcanzable desde s (sino $\delta(s, v) = \infty \geq d[v]$).

Sea u un vértice que precede inmediatamente a v en un camino más corto de s a v (u existe porque $v \neq s$): $s \rightsquigarrow u \to v$

Búsqueda en amplitud: Correctitud

Demostrando: (i) $d[v] = \delta(s, v)$ para todo $v \in V$

Sea u un vértice que precede inmediatamente a v en un camino más corto de s a v (u existe porque $v \neq s$): $s \rightsquigarrow u \to v$

Por elección de v, $\delta(s, u) = d[u]$

Por elección de u, $\delta(s, v) = \delta(s, u) + 1$

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Consideremos el color de v cuando u es decolado (línea 15):

1. v *es BLANCO:*

La línea 19 coloca $d[v] = d[u] + 1$ (contradicción)

Búsqueda en amplitud: Correctitud

Demostrando: (i) $d[v] = \delta(s, v)$ para todo $v \in V$

Sea u un vértice que precede inmediatamente a v en un camino más corto de s a v (u existe porque $v \neq s$): $s \rightsquigarrow u \to v$

Por elección de v, $\delta(s, u) = d[u]$

Por elección de u, $\delta(s, v) = \delta(s, u) + 1$

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Consideremos el color de v cuando u es decolado (línea 15):

2. v *es NEGRO:*

v fue decolado anteriormente y por lo tanto encolado antes que u

Por el Corolario, $d[v] \leq d[u]$ (contradicción)

Búsqueda en amplitud: Correctitud

Demostrando: (i) $d[v] = \delta(s, v)$ para todo $v \in V$

Sea u un vértice que precede inmediatamente a v en un camino más corto de s a v (u existe porque $v \neq s$): $s \rightsquigarrow u \to v$

Por elección de v, $\delta(s, u) = d[u]$

Por elección de u, $\delta(s, v) = \delta(s, u) + 1$

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Consideremos el color de v cuando u es decolado (línea 15):

3. v *es GRIS:*

v fue pintado GRIS al decolar otro vértice w (decolado antes que u)

Por el Corolario, $d[w] \leq d[u]$ y entonces $d[v] = d[w] + 1 \leq d[u] + 1$ (contradicción)

Búsqueda en amplitud: Correctitud

Demostrando: (i) $d[v] = \delta(s, v)$ para todo $v \in V$

Sea u un vértice que precede inmediatamente a v en un camino más corto de s a v (u existe porque $v \neq s$): $s \rightsquigarrow u \rightarrow v$

Por elección de v, $\delta(s, u) = d[u]$

Por elección de u, $\delta(s, v) = \delta(s, u) + 1$

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

Consideremos el color de v cuando u es decolado (línea 15):
En los tres casos alcanzamos una contradiccón. Entonces, no existe vértice v con $d[v] \neq \delta(s, v)$

Búsqueda en amplitud: Correctitud

(ii) para cada $v \in V$ con $v \neq s$, existe un camino más corto de s a v, que contiene la arista $(\pi[v], v)$

Si $\pi[v] = u$, entonces $\delta(s, v) = d[v] = d[u] + 1 = \delta(s, u) + 1$

Por el lema de caminos más cortos, un camino más corto de s a u **extendido** con la arista $(\pi[v], v) = (u, v)$ es un camino más corto de s a v □

Búsqueda en amplitud: Árbol breadth-first

Al terminar BFS, definimos el **árbol de predecesores** $G_\pi = (V_\pi, E_\pi)$:

- $V_\pi = \{v \in V : \pi[v] \neq \text{null}\} \cup \{s\}$
- $E_\pi = \{(\pi[v], v) : v \in V_\pi \setminus \{s\}\}$

V_π es $\{s\}$ unido a todos los vértices descubiertos por BFS

E_π son todos las aristas $(\pi[v], v)$ para $\pi[v] \neq \text{null}$

Búsqueda en amplitud: Árbol breadth-first

$V_\pi = \{v \in V : \pi[v] \neq \text{null}\} \cup \{s\}$; $E_\pi = \{(\pi[v], v) : v \in V_\pi \setminus \{s\}\}$

Lema

El grafo $G_\pi = (V_\pi, E_\pi)$ construido por BFS es un árbol breadth-first

Prueba: tenemos que probar dos cosas:

- G_π es un árbol
- el único camino de s a $v \in V_\pi$ es un camino más corto en G

1 G_π *es un árbol:* claramente $|E_\pi| = |V_\pi| - 1$. Por otro lado, G_π es conectado ya que todo vértice en V_π es alcanzable desde s

Por lo tanto, G_π es un árbol

Búsqueda en amplitud: Árbol breadth-first

$$V_\pi = \{v \in V : \pi[v] \neq \text{null}\} \cup \{s\} \quad ; \quad E_\pi = \{(\pi[v], v) : v \in V_\pi \setminus \{s\}\}$$

Lema

El grafo $G_\pi = (V_\pi, E_\pi)$ construido por BFS es un árbol breadth-first

Prueba: tenemos que probar dos cosas:

- G_π es un árbol
- el único camino de s a $v \in V_\pi$ es un camino más corto en G

2 *el único camino de s a $v \in V_\pi$ es un camino más corto en G:*

Sea $(s, v_1, v_2, \ldots, v_n)$ un camino en G_π. Utilizando el Teorema de Correctitud repetidamente, se muestra por inducción que el camino $(s, v_1, \ldots, v_i)$ es un camino más corto en G, para $i = 1, \ldots, n$ (ejercicio) $\square$