Búsqueda en profundidad

Guillermo Palma

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

Plan

1. Algoritmo de búsqueda en profundidad
2. Propiedades de la búsqueda en profundidad
3. Clasificación de los lados en un bosque de la búsqueda en profundidad

Algoritmo de búsqueda en profundidad

Sobre la búsqueda en profundidad o Depth-first search (DFS)

- Recorre los lados de un grafo desde los vértices del grafo hasta lo más profundo posible.
- Se recorre los lados del vértice más recientemente visitado que no haya sido visitado antes.
- Cuando no se puede recorrer más allá de un vértice, se escoge el anteriormente visitado (*backtrack*).
- Se aplica a todos los vértices del grafo.
- El algoritmo termina cuando visita a todos los vértices.
- El grafo subgrafo predecesor que genera puede tener varios árboles.
- Al principio todos los vértices son coloreados de blanco y cuando de termina de visitarlos se colorea de negro.
- Los vértices tienen dos tiempos:
  1. un tiempo inicial cuando fueron descubiertos.
  2. un tiempo final cuando terminan de ser examinado.
- Hay un versión que es similar a BFS pero usa una pila en vez de cola.

Subgrafo predecesor de la búsqueda en profundidad

Definición

Sea $G(V, E)$ un digrafo o un grafo no dirigido, en el se ejecuta el algoritmo de búsqueda en profundidad. Se define el subrafo predecesor $G(V_{pred}, E_{pred})$ generado por la búsqueda en profundidad como:

$$G_{pred} = (V, E_{pred})$$
$$E_{pred} = \{(v_{pred}, v) : v \in V \wedge v_{pred} \neq NIL\}$$

- Se genera varios árboles (*depth-first trees*).
- Los árboles conforman un bosque (*depth-first forest*).
- E_{pred} son los lados del árbol (*tree edges*).

Algoritmo de búsqueda en profundidad

Procedimiento DFS(G =(V, E))

```
inicio
    para cada (v ∈ V) hacer v.color ← BLANCO; v.pred ← NIL ;
    tiempo ← 0 ; /* Variable global */
    para cada v ∈ V hacer
        si (v.color = BLANCO) entonces dfsVisit(G, v) ;
```

Procedimiento dfsVisit(G, v)

```
inicio
    tiempo ← tiempo + 1 ; /* se empieza a explorar v */
    v.d ← tiempo ; /* tiempo inicial */
    v.color ← GRIS ;
    para cada (u ∈ G.adyacentes[v]) hacer
        si (u.color = BLANCO) entonces
            u.pred ← v ;
            dfsVisit(G, u)
    v.color ← NEGRO ; /* se termina de explorar v */
    tiempo ← tiempo + 1 ;
    v.f ← tiempo ; /* tiempo final */
```

Ejemplo de la búsqueda en profundidad

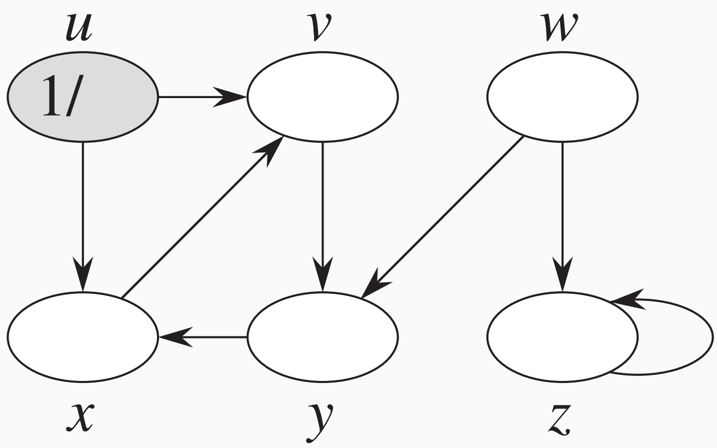


Figura 1: Se alcanza el vértice u. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

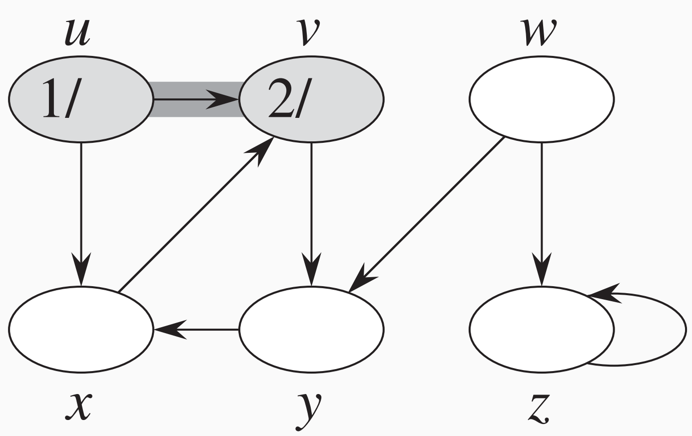


Figura 2: Se alcanza el vértice v. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

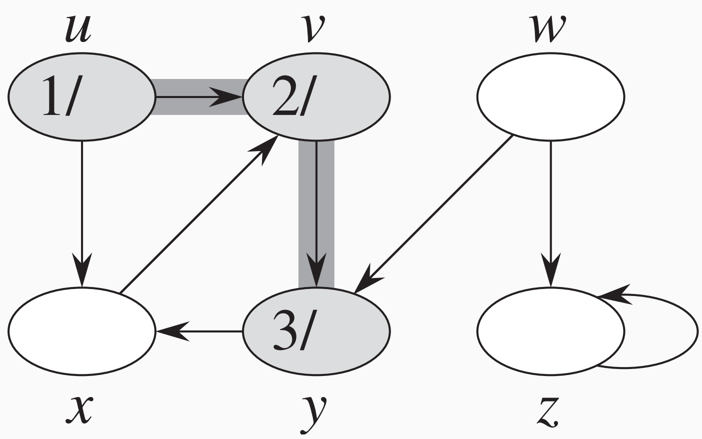


Figura 3: Se alcanza el vértice y. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

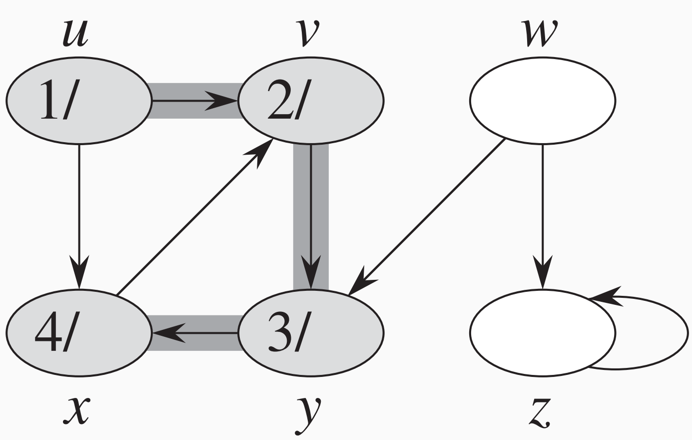


Figura 4: Se alcanza el vértice x. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

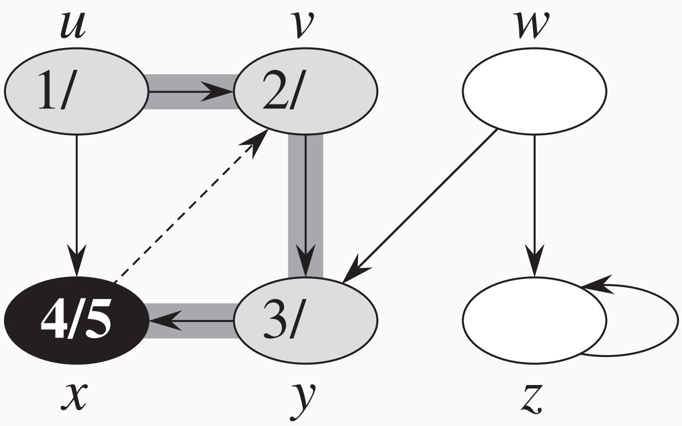


Figura 5: Se cierra el vértice x. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

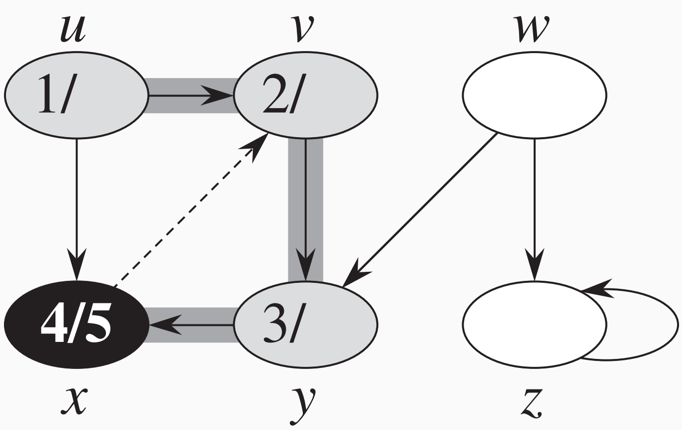


Figura 6: Se cierra el vértice x. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

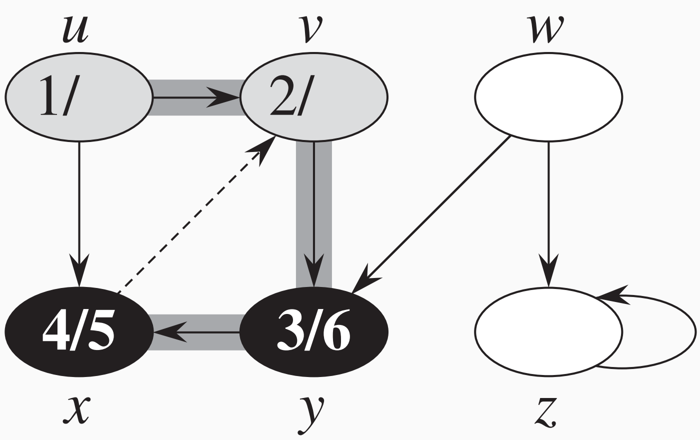


Figura 7: Se cierra el vértice y. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

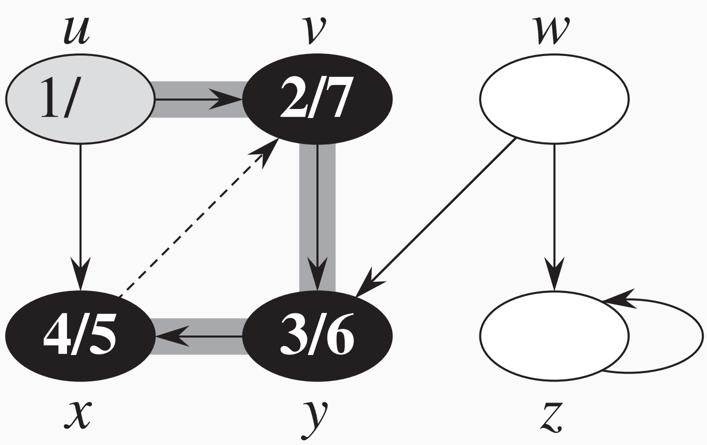


Figura 8: Se cierra el vértice v. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

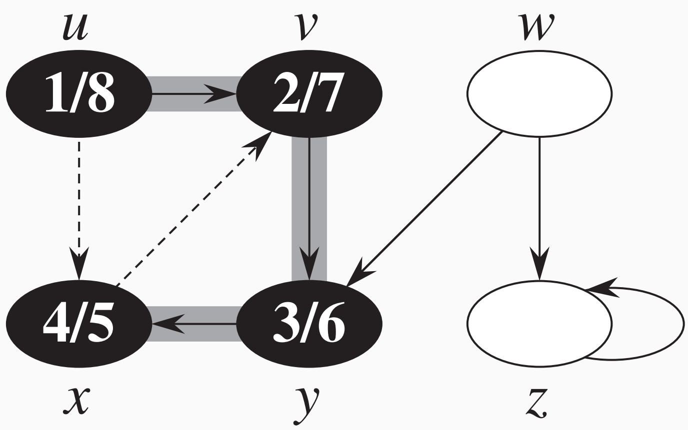


Figura 9: Se cierra el vértice u. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

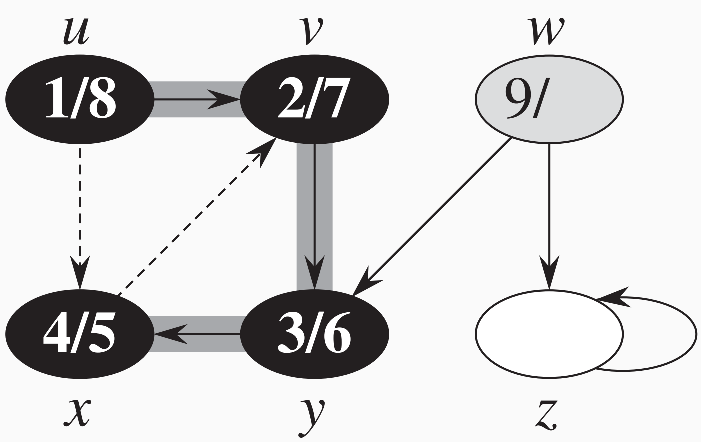


Figura 10: Se alcanza el vértice w. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

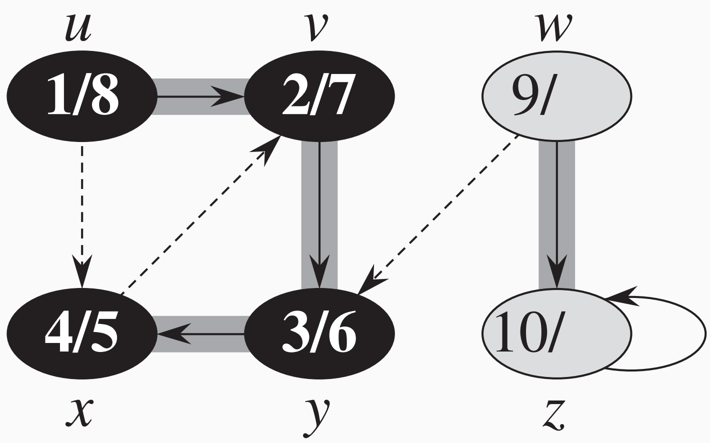


Figura 11: Se alcanza el vértice z. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

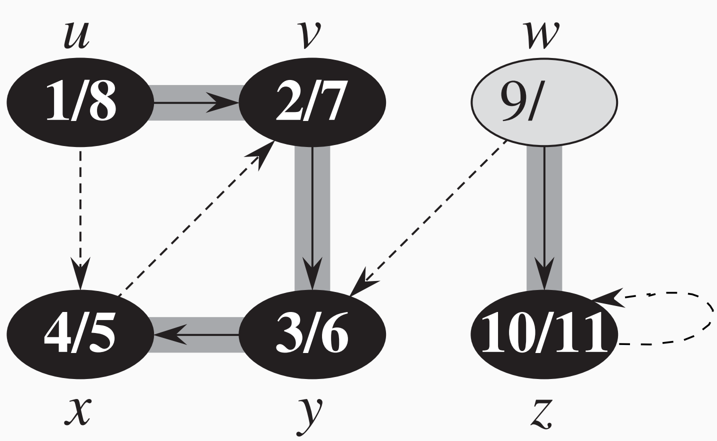


Figura 12: Se cierra el vértice z. Fuente [1].

Ejemplo de la búsqueda en profundidad, continuación

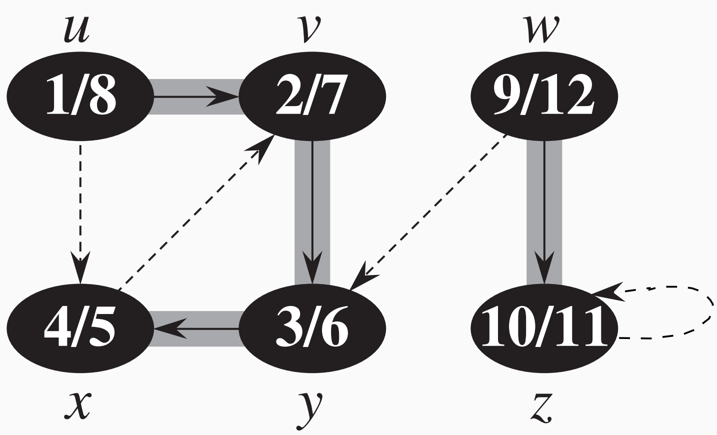


Figura 13: Se cierra el vértice w y se termina la búsqueda en profundidad. Fuente [1].

Análisis de la búsqueda en profundidad

- En el procedimiento `DFS` las líneas 2 y 4 son $\Theta(|V|)$.
- Desde `DFS` se llama al procedimiento `dfsVisit` en la línea 5 para todos los vértices.
- Un vértice v entra `dfsVisit` solo cuando es color blanco.
- Una vez que un vértice entra a `dfsVisit` se colorea de gris.
- En consecuencia, cada vértice entra una sola vez a `dfsVisit`.
- En `dfsVisit` cada vértice v, ejecutará en el peor caso las líneas 5-8 tantos adyacentes tenga el vértice.
- Si todos vértice $v \in V$ en la línea 4 de `DFS`, ejecutan `dfsVisit` en el peor caso $O(|G.adyacentes[v]|)$, entonces se la operación es $\Theta(|E|)$.
- Por lo tanto, el tiempo de DFS es $\Theta(|V| + |E|)$.

Propiedades de la búsqueda en profundidad

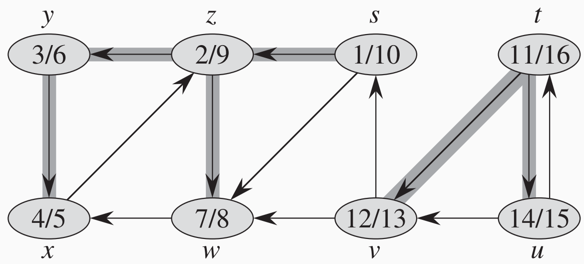
Paréntesis de la búsqueda en profundidad

Teorema del paréntesis

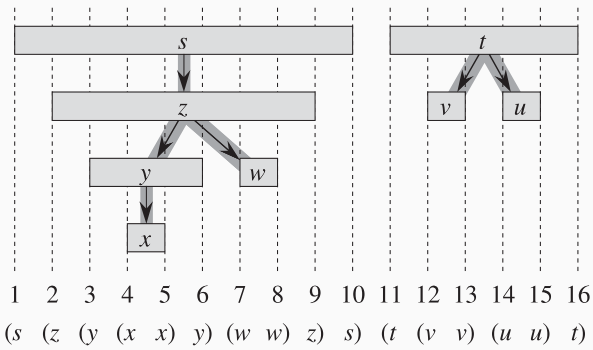
En una búsqueda en profundidad de un grafo $G = (V, E)$, digrafo o no dirigido, para cualquier para de vértices $u, v \in V$, se tiene que alguna de estas tres opciones se cumple:

- $[u.d, u.f]$ y $[v.d, v.f]$ son disjuntos y se tiene que v y u no son descendientes uno de otro en el bosque de la búsqueda en profundidad.
- $[u.d, u.f]$ es contenido completamente en $[v.d, v.f]$ y se tiene que u es un descendiente de v en el bosque de la búsqueda en profundidad.
- $[v.d, v.f]$ es contenido completamente en $[u.d, u.f]$ y se tiene que v es un descendiente de u en el bosque de la búsqueda en profundidad.

Ejemplo de los paréntesis generados por DFS



(a) Bosque con dos árboles obtenidos por DFS.



(b) Paréntesis de los árboles de DFS.

Figura 14: Resultado de DFS y la paréntesis de los vértices. Fuente [1].

Paréntesis de la búsqueda en profundidad, cont.

Prueba del Teorema del paréntesis

Caso 1 $u.d < v.d$: se tiene que u fue descubierto antes que v

Subcaso 1 $v.d < u.f$:

- se tiene que v fue descubierto cuando u era todavía gris,
- entonces cuando u finaliza después de v, esto es $v.f < u.f$.
- En consecuencia v es descendiente de u.
- Se tiene que $u.d < v.d < v.f < u.f$.
- Por lo tanto $[v.d, v.f]$ es contenido completamente en $[u.d, u.f]$ y v es un descendiente de u.

Subcaso 2 $v.d > u.f$:

- Se tiene que u finaliza antes que v sea descubierto.
- En consecuencia, $v.f > v.d > u.f > u.d$, entonces v no es descendiente de u.
- Por lo tanto, $[v.d, v.f]$ y $[u.d, u.f]$ son disjuntos y se tiene que v y u no son descendientes uno de otro.

Caso 2 $u.d > v.d$: Análogo con el caso 1 solo se intercambian u y v. □

Intervalos de descendientes de la búsqueda en profundidad

Corolario 1 (anidamiento de intervalos de descendientes)

Un vértice v es descendiente de un vértice u en un bosque generado por la búsqueda en profundidad en un grafo G, sí solo si
$u.d < v.d < v.f < u.f$.

Prueba

Por el Teorema del paréntesis se cumple que v es descendiente de un vértice u.

Camino blanco desde un vértice hasta otro en DFS

Teorema del camino blanco

En un bosque generado por la búsqueda en profundidad en un grafo G, un vértice v es descendiente de un vértice u si solo si existe en el tiempo $u.d$ en el que u es descubierto, un camino de vértices blancos desde u hasta v.

Camino blanco desde un vértice hasta otro en DFS, cont.

Prueba del Teorema del camino blanco

$\Rightarrow$

- Si $u = v$ se tiene que u es blanco cuando es descubierto.
- Suponemos que v es descendiente de u.
- Por Corolario 1 se tiene que $u.d < v.d$ y v es blanco cuando es descubierto,
- Se tiene que v puede ser cualquier vértice en el camino de u hasta v.
- Por lo tanto, cualquier camino simple desde u hasta v tiene todos los vértices en blanco.

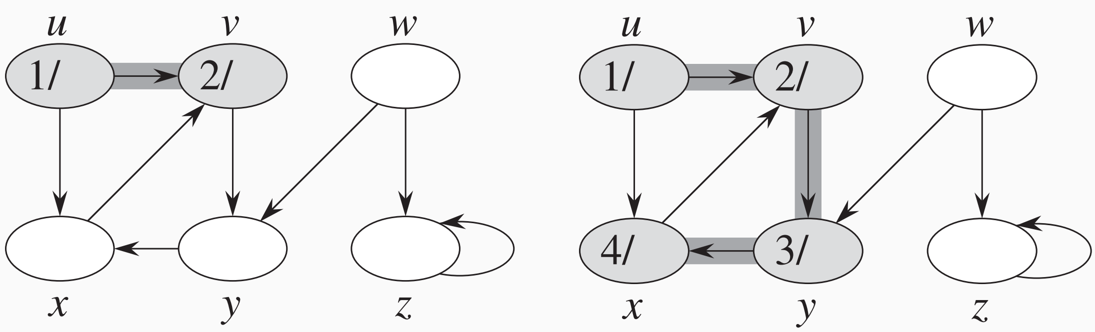
Camino blanco desde un vértice hasta otro en DFS

Continuación Prueba del Teorema del camino blanco

$\Leftarrow$

- Suponemos que hay un camino blanco desde u hasta v, pero v no es descendiente de u.
- Sea w el predecesor de v en el camino y sea w descendiente de u en el camino (puede ser $w = u$).
- Por Corolario 1 $w.f \leq u.f$.
- Se tiene que v debe ser descubierto después de u pero antes w es finalizado.
- $u.d < v.d < w.f \leq u.f$.
- Por Teorema del paréntesis el intervalo $[v.d, v.f]$ está contenido completamente en $[u.d, u.f]$.
- Por lo tanto, por Corolario 1 v debe ser descendiente de u.

Ejemplo de camino blanco en DFS



(a) Se alcanza el vértice v.　　(b) Se alcanza el vértice x.

Figura 15: Ejemplo de un camino blanco en el que se observa que el vértice x es descendiente del vértice v. Fuente [1].

Clasificación de los lados en un bosque de la búsqueda en profundidad

Clasificación de los lados en un bosque de DFS

1. **Tree edges**: lados del bosque de DFS.
2. **Back edges**: lado (u, v) desde un vértice v hasta su ancestro u en un árbol DFS.
3. **Forward edge**: lado (u, v) que no son del árbol DFS y en el que se conecta a un vértice u con su descendiente v en un árbol DFS.
4. **Cross edges**: cualquiera de los otros lados.

Ejemplo de los tipos de lados de DFS

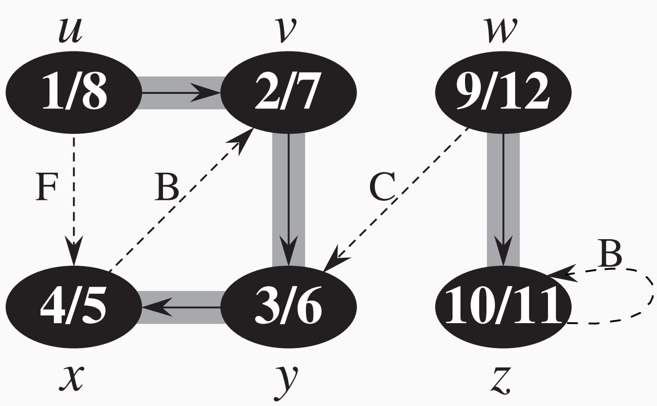


Figura 16: Bosque resultante de la ejecución de DFS en la Figura 13, con los lados clasificados. Fuente [1].

Ejemplo del bosque generado por DFS y sus lados

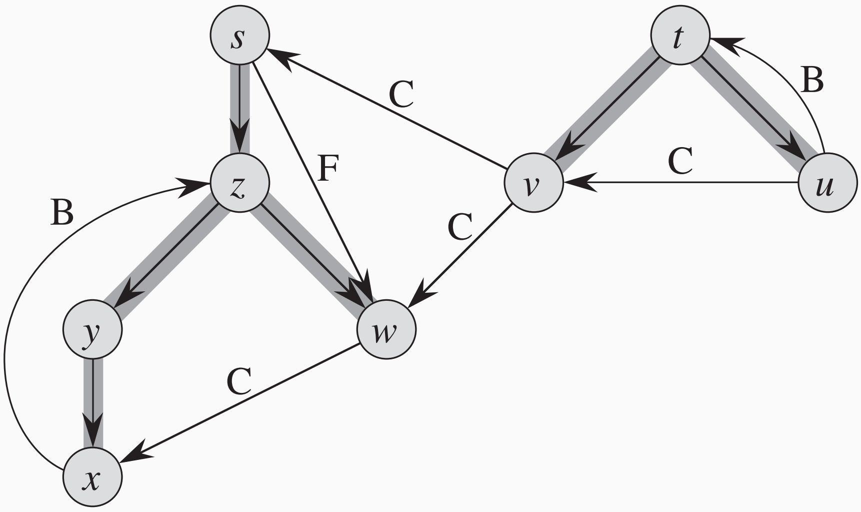


Figura 17: Bosque resultante de la ejecución de DFS en la Figura 14(a), con los lados clasificados. Fuente [1].

Lados generados por DFS en grafo no dirigido

Teorema

En un bosque generado por DFS en un grafo no dirigido G, cada lado de G o es *tree edge* o es un *back edge*.

Referencias

[1] T. Cormen, C. Leirserson, R. Rivest, and C. Stein.
Introduction to Algorithms.
McGraw Hill, 3ra edition, 2009.