Computación y Estructuras Discretas I

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes

UNIVERSIDAD ICESI

2024-2

Agenda del día

Introducción

¿Qué es un recorrido?

Introducción

¿Qué es un recorrido?

Es un procedimiento sistemático de exploración de un grafo mediante la visita a todos sus vértices y aristas.

Introducción

¿Qué es un recorrido?

Es un procedimiento sistemático de exploración de un grafo mediante la visita a todos sus vértices y aristas.

¿Cuándo un recorrido es eficiente?

Introducción

¿Qué es un recorrido?

Es un procedimiento sistemático de exploración de un grafo mediante la visita a todos sus vértices y aristas.

¿Cuándo un recorrido es eficiente?

Un recorrido es eficiente si visita todos los vértices y aristas en un tiempo proporcional a su número, esto es, en tiempo lineal.

Introducción

¿Cuál es el interés de las búsquedas en grafos?

Introducción

¿Cuál es el interés de las búsquedas en grafos?

- Recorridos parciales en grafos infinitos o tan grandes que resultan inmanejables.

Introducción

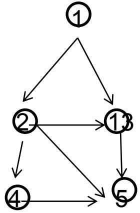
¿Cuál es el interés de las búsquedas en grafos?

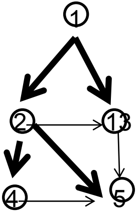
- Recorridos parciales en grafos infinitos o tan grandes que resultan inmanejables.
- Buscar un elemento en un grafo a partir de un origen.

Introducción

¿Cuál es el interés de las búsquedas en grafos?

- Recorridos parciales en grafos infinitos o tan grandes que resultan inmanejables.
- Buscar un elemento en un grafo a partir de un origen.





Introducción

¿Cuáles son los tipos de búsquedas en grafos?

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información
  - No utiliza información sobre el problema.

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información
  - No utiliza información sobre el problema.
  - Normalmente, se realiza una búsqueda exhaustiva.

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información
  - No utiliza información sobre el problema.
  - Normalmente, se realiza una búsqueda exhaustiva.
- Búsqueda heurística

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información
  - No utiliza información sobre el problema.
  - Normalmente, se realiza una búsqueda exhaustiva.
- Búsqueda heurística
  - Utiliza información sobre el problema como costos, etc.

Introducción

¿Cuáles son los tipos de búsquedas en grafos?

- Búsqueda ciega o sin información
  - No utiliza información sobre el problema.
  - Normalmente, se realiza una búsqueda exhaustiva.
- Búsqueda heurística
  - Utiliza información sobre el problema como costos, etc.
  - Se posee información muy valiosa para orientar la búsqueda

Introducción

¿En que consiste una búsqueda ciega?

Introducción

¿En que consiste una búsqueda ciega?

- La búsqueda consiste en escoger uno de los nodos posibles.

Introducción

¿En que consiste una búsqueda ciega?

- La búsqueda consiste en escoger uno de los nodos posibles.
- Si hay varios nodos que se pueden expandir, la elección de cual se expande primero se hace según una estrategia de búsqueda.

Introducción

¿En que consiste una búsqueda ciega?

- La búsqueda consiste en escoger uno de los nodos posibles.
- Si hay varios nodos que se pueden expandir, la elección de cual se expande primero se hace según una estrategia de búsqueda.
- El proceso de búsqueda se concibe como la construcción de un recorrido del grafo.

Introducción

¿En que consiste una búsqueda ciega?

- La búsqueda consiste en escoger uno de los nodos posibles.
- Si hay varios nodos que se pueden expandir, la elección de cual se expande primero se hace según una estrategia de búsqueda.
- El proceso de búsqueda se concibe como la construcción de un recorrido del grafo.
- Las técnicas se diferencian en el orden en que expanden los nodos.

Introducción

¿Cómo podría ser la estrategia de búsqueda?

Introducción

¿Cómo podría ser la estrategia de búsqueda?

- Por extensión o amplitud

Introducción

¿Cómo podría ser la estrategia de búsqueda?

- Por extensión o amplitud
  - Origen, sus vecinos, etc.

Introducción

¿Cómo podría ser la estrategia de búsqueda?

- Por extensión o amplitud
  - Origen, sus vecinos, etc.
- Por profundidad

Introducción

¿Cómo podría ser la estrategia de búsqueda?

- Por extensión o amplitud
  - Origen, sus vecinos, etc.
- Por profundidad
  - Expande un nodo hasta que no se pueda más y regresa.

Agenda del día

1. **Búsquedas en grafos**
   - Introducción
   - BFS
   - Ejercicios
   - DFS
   - Ejercicios

BFS

¿Qué es el BFS?

BFS

¿Qué es el BFS?

- El BFS, o búsqueda en amplitud (anchura) es uno de los algoritmos más simples de búsqueda en grafos.

BFS

¿Qué es el BFS?

- El BFS, o búsqueda en amplitud (anchura) es uno de los algoritmos más simples de búsqueda en grafos.
- Es el prototipo de muchos otros algoritmos importantes sobre grafos.

BFS

¿Qué es el BFS?

- El BFS, o búsqueda en amplitud (anchura) es uno de los algoritmos más simples de búsqueda en grafos.
- Es el prototipo de muchos otros algoritmos importantes sobre grafos.

¿Intuitivamente qué hace el BFS?

BFS

¿Qué es el BFS?

- El BFS, o búsqueda en amplitud (anchura) es uno de los algoritmos más simples de búsqueda en grafos.
- Es el prototipo de muchos otros algoritmos importantes sobre grafos.

¿Intuitivamente qué hace el BFS?

- Se comienza con un nodo origen y se exploran todos los vecinos de este nodo.

BFS

¿Qué es el BFS?

- El BFS, o búsqueda en amplitud (anchura) es uno de los algoritmos más simples de búsqueda en grafos.
- Es el prototipo de muchos otros algoritmos importantes sobre grafos.

¿Intuitivamente qué hace el BFS?

- Se comienza con un nodo origen y se exploran todos los vecinos de este nodo.
- A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo.

BFS

¿Y un poco más formal?

BFS

¿Y un poco más formal?

- Dado un grafo $G = (V, E)$ y un vértice origen s, el BFS sistemáticamente explora las aristas de G para descubrir cada vértice que es alcanzable desde s.

BFS

¿Y un poco más formal?

- Dado un grafo $G = (V, E)$ y un vértice origen s, el BFS sistemáticamente explora las aristas de G para descubrir cada vértice que es alcanzable desde s.
- Computa la distancia (menor cantidad de vértices) de s a cada vértice alcanzable.

BFS

¿Y un poco más formal?

- Dado un grafo $G = (V, E)$ y un vértice origen s, el BFS sistemáticamente explora las aristas de G para descubrir cada vértice que es alcanzable desde s.
- Computa la distancia (menor cantidad de vértices) de s a cada vértice alcanzable.
- También produce un árbol BF con s como raíz y todos los vértices alcanzables desde s.

BFS

¿Y un poco más formal?

- Dado un grafo $G = (V, E)$ y un vértice origen s, el BFS sistemáticamente explora las aristas de G para descubrir cada vértice que es alcanzable desde s.
- Computa la distancia (menor cantidad de vértices) de s a cada vértice alcanzable.
- También produce un árbol BF con s como raíz y todos los vértices alcanzables desde s.
- Funciona para grafos dirigidos y no dirigidos.

BFS

¿Qué información se puede obtener a partir del árbol BF?

BFS

¿Qué información se puede obtener a partir del árbol BF?

- Para cada vértice *v*, alcanzable desde *s*, el camino simple de *s* a *v* en el árbol BF resultante, corresponde al camino más corto (menor cantidad de aristas) de *s* a *v* en el grafo.

BFS

¿Por qué búsqueda por anchura?

BFS

¿Por qué búsqueda por anchura?

- Este algoritmo de búsqueda toma este nombre ya que expande uniformemente la frontera entre los vértices descubiertos y no descubiertos a través de la anchura de su frontera.

BFS

¿Por qué búsqueda por anchura?

- Este algoritmo de búsqueda toma este nombre ya que expande uniformemente la frontera entre los vértices descubiertos y no descubiertos a través de la anchura de su frontera.
- El algoritmo descubre todos los vértices a una distancia k desde s antes de descubrir algún vértice a una distancia $k + 1$.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.
- Todos los vértices comienzan como blancos y posteriormente se pueden tornar grises y luego negros.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.
- Todos los vértices comienzan como blancos y posteriormente se pueden tornar grises y luego negros.
- Un vértice **se descubre** por primera vez si se encuentra durante la búsqueda y en ese momento deja de ser blanco.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.
- Todos los vértices comienzan como blancos y posteriormente se pueden tornar grises y luego negros.
- Un vértice **se descubre** por primera vez si se encuentra durante la búsqueda y en ese momento deja de ser blanco.
    - Vértices negros o grises ya han sido descubiertos pero BFS los distingue para garantizar que la búsqueda proceda por anchura.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.
- Todos los vértices comienzan como blancos y posteriormente se pueden tornar grises y luego negros.
- Un vértice **se descubre** por primera vez si se encuentra durante la búsqueda y en ese momento deja de ser blanco.
  - Vértices negros o grises ya han sido descubiertos pero BFS los distingue para garantizar que la búsqueda proceda por anchura.
  - Si $(u, v) \in E$ y u es negro, entonces v es negro o gris. Es decir, todos los vértices adyacentes a vértices negros ya han sido descubiertos.

BFS

¿Cómo mantiene el BFS el registro de su progreso?

- BFS colorea cada vértice de blanco, gris o negro.
- Todos los vértices comienzan como blancos y posteriormente se pueden tornar grises y luego negros.
- Un vértice **se descubre** por primera vez si se encuentra durante la búsqueda y en ese momento deja de ser blanco.
  - Vértices negros o grises ya han sido descubiertos pero BFS los distingue para garantizar que la búsqueda proceda por anchura.
  - Si $(u, v) \in E$ y u es negro, entonces v es negro o gris. Es decir, todos los vértices adyacentes a vértices negros ya han sido descubiertos.
  - Los vértices grises pueden tener algunos vértices adyacentes blancos. Estos representan la frontera entre vértices descubiertos y no descubiertos.

BFS

¿Cómo construye el BFS el árbol BF?

BFS

¿Cómo construye el BFS el árbol BF?

- Inicialmente BFS construye el árbol BF con sólo la raíz que sería el vértice origen s.

BFS

¿Cómo construye el BFS el árbol BF?

- Inicialmente BFS construye el árbol BF con sólo la raíz que sería el vértice origen s.
- Cuando la búsqueda descubre un vértice blanco v mientras escanea la lista de adyacencia de un vértice u ya descubierto, el vértice v y la arista (u, v) se adicionan al árbol.

BFS

¿Cómo construye el BFS el árbol BF?

- Inicialmente BFS construye el árbol BF con sólo la raíz que sería el vértice origen s.
- Cuando la búsqueda descubre un vértice blanco v mientras escanea la lista de adyacencia de un vértice u ya descubierto, el vértice v y la arista (u, v) se adicionan al árbol.
- Se dice que u es el predecesor o padre de v en el árbol BF.

BFS

¿Cómo construye el BFS el árbol BF?

- Inicialmente BFS construye el árbol BF con sólo la raíz que sería el vértice origen s.
- Cuando la búsqueda descubre un vértice blanco v mientras escanea la lista de adyacencia de un vértice u ya descubierto, el vértice v y la arista (u, v) se adicionan al árbol.
- Se dice que u es el predecesor o padre de v en el árbol BF.
- Como un vértice se descubre una vez como máximo, entonces tiene un padre, como máximo.

BFS

¿Cómo construye el BFS el árbol BF?

- Inicialmente BFS construye el árbol BF con sólo la raíz que sería el vértice origen s.
- Cuando la búsqueda descubre un vértice blanco v mientras escanea la lista de adyacencia de un vértice u ya descubierto, el vértice v y la arista (u, v) se adicionan al árbol.
- Se dice que u es el predecesor o padre de v en el árbol BF.
- Como un vértice se descubre una vez como máximo, entonces tiene un padre, como máximo.
- Las relaciones de ancestro y descendiente se definen relativo a la raíz s.

BFS

¿Qué asumimos antes de presentar el BFS?

BFS

¿Qué asumimos antes de presentar el BFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.

BFS

¿Qué asumimos antes de presentar el BFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.

BFS

¿Qué asumimos antes de presentar el BFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
    - Para cada vértice $u \in V$ el color se guarda en el atributo *u.color*, el predecesor en el atributo $u.\pi$ y la distancia entre s y u en $u.d$.

BFS

¿Qué asumimos antes de presentar el BFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo $u.color$, el predecesor en el atributo $u.\pi$ y la distancia entre s y u en $u.d$.
- El algoritmo utiliza una cola Q para manejar el conjunto de vértices grises.

BFS

¿Cómo sería el pseudocódigo para el BFS?

BFS

¿Cómo sería el pseudocódigo para el BFS?

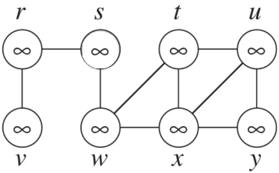
```
BFS(G, s)
for each vertex u ∈ G.V − {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = GRAY
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each v ∈ G.Adj[u]
        if v.color == WHITE
            v.color = GRAY
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)
    u.color = BLACK
```

BFS

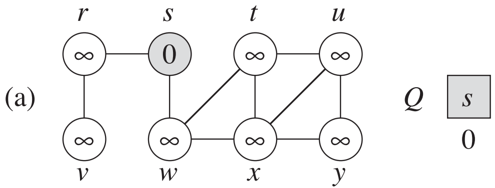
¿Cómo sería el funcionamiento del BFS en un grafo no dirigido?

BFS

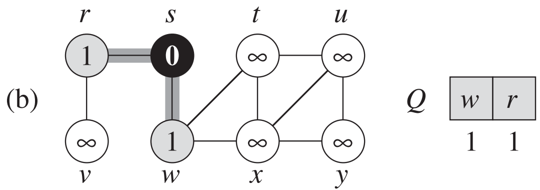
¿Cómo sería el funcionamiento del BFS en un grafo no dirigido?



BFS

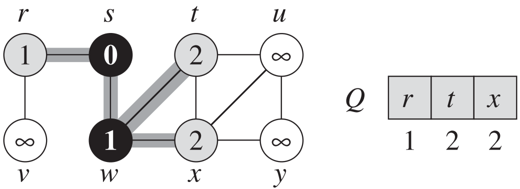


BFS

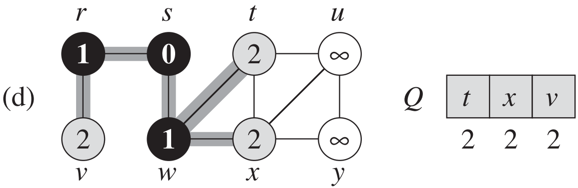


BFS

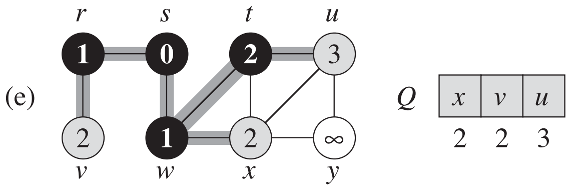
(c)



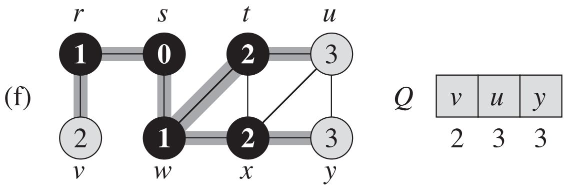
BFS



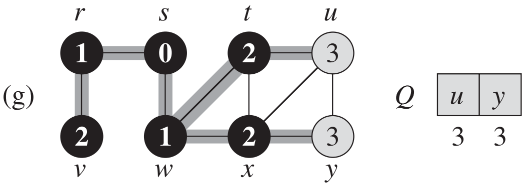
BFS



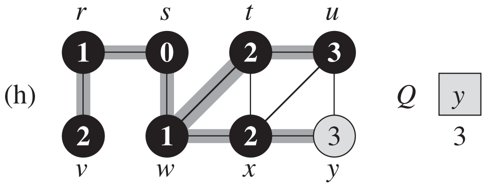
BFS



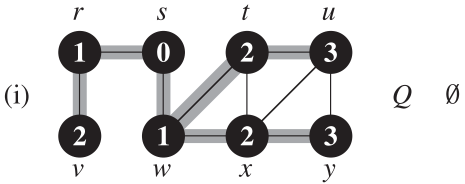
BFS



BFS



BFS



BFS

¿Cuál es la complejidad temporal del BFS?

BFS

¿Cuál es la complejidad temporal del BFS?

- $O(V + E)$.

BFS

¿Qué es un árbol BF?

BFS

¿Qué es un árbol BF?

- El BFS construye un árbol BF a medida que busca en el grafo.

BFS

¿Qué es un árbol BF?

- El BFS construye un árbol BF a medida que busca en el grafo.
- Este árbol corresponde a los atributos π.

BFS

¿Qué es un árbol BF?

- El BFS construye un árbol BF a medida que busca en el grafo.
- Este árbol corresponde a los atributos π.
- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V_\pi, E_\pi)$ donde

BFS

¿Qué es un árbol BF?

- El BFS construye un árbol BF a medida que busca en el grafo.
- Este árbol corresponde a los atributos π.
- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V_\pi, E_\pi)$ donde

$$V_\pi = \{v \in V \mid v.\pi \neq NIL\} \cup \{s\} \text{ y}$$
$$E_\pi = \{(v.\pi, v) \mid v \in V_\pi - \{s\}\}$$

BFS

¿Qué es un árbol BF?

- El BFS construye un árbol BF a medida que busca en el grafo.
- Este árbol corresponde a los atributos π.
- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V_\pi, E_\pi)$ donde

$$V_\pi = \{v \in V \mid v.\pi \neq NIL\} \cup \{s\} \text{ y}$$
$$E_\pi = \{(v.\pi, v) \mid v \in V_\pi - \{s\}\}$$

- El grafo de predecesores G_π es un árbol BF si V_π consiste de los vértices alcanzables desde s y para todo $v \in V_\pi$, el subgrafo G_π contiene un único camino simple de s a v que también es el camino más corto de s a v en G.

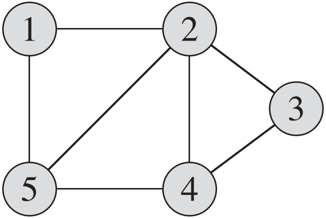
Agenda del día

1. **Búsquedas en grafos**
   - Introducción
   - BFS
   - Ejercicios
   - DFS
   - Ejercicios

Ejercicios

Ejercicio

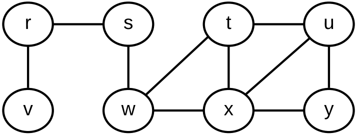
Muestre los valores d y π que resultan de correr el BFS sobre el siguiente grafo no dirigido cuando el origen es el vértice 3.



Ejercicios

Ejercicio

Muestre los valores d y π que resultan de correr el BFS sobre el siguiente grafo no dirigido cuando el origen es el vértice u.



Agenda del día

DFS

¿Qué es el DFS?

DFS

¿Qué es el DFS?

- El DFS, o búsqueda en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme.

DFS

¿Qué es el DFS?

- El DFS, o búsqueda en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme.
- Expande un nodo hasta que no se pueda más y regresa.

DFS

¿Qué es el DFS?

- El DFS, o búsqueda en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme.
- Expande un nodo hasta que no se pueda más y regresa.

¿Intuitivamente qué hace el DFS?

DFS

¿Qué es el DFS?

- El DFS, o búsqueda en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme.
- Expande un nodo hasta que no se pueda más y regresa.

¿Intuitivamente qué hace el DFS?

- Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto.

DFS

¿Qué es el DFS?

- El DFS, o búsqueda en profundidad es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme.
- Expande un nodo hasta que no se pueda más y regresa.

¿Intuitivamente qué hace el DFS?

- Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto.
- Cuando ya no quedan más nodos que visitar en dicho camino, regresa, de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

DFS

¿Cuál es la estrategia que sigue el DFS?

DFS

¿Cuál es la estrategia que sigue el DFS?

- Como su nombre lo indica, busca cada vez más profundo en el grafo, mientras sea posible.

DFS

¿Cuál es la estrategia que sigue el DFS?

- Como su nombre lo indica, busca cada vez más profundo en el grafo, mientras sea posible.
- Explora las aristas desde el vértice v que ha sido descubierto más recientemente y que aún tiene aristas sin explorar.

DFS

¿Cuál es la estrategia que sigue el DFS?

- Como su nombre lo indica, busca cada vez más profundo en el grafo, mientras sea posible.
- Explora las aristas desde el vértice v que ha sido descubierto más recientemente y que aún tiene aristas sin explorar.
- Cuando todas las aristas de v han sido exploradas, la búsqueda se devuelve para explorar las aristas que salen de donde fue descubierto v.

DFS

¿Cuál es la estrategia que sigue el DFS?

- Como su nombre lo indica, busca cada vez más profundo en el grafo, mientras sea posible.
- Explora las aristas desde el vértice v que ha sido descubierto más recientemente y que aún tiene aristas sin explorar.
- Cuando todas las aristas de v han sido exploradas, la búsqueda se devuelve para explorar las aristas que salen de donde fue descubierto v.
- Este proceso continúa hasta que se descubran todos los vértices alcanzables desde el vértice original.

DFS

¿Cuál es la estrategia que sigue el DFS?

- Como su nombre lo indica, busca cada vez más profundo en el grafo, mientras sea posible.
- Explora las aristas desde el vértice v que ha sido descubierto más recientemente y que aún tiene aristas sin explorar.
- Cuando todas las aristas de v han sido exploradas, la búsqueda se devuelve para explorar las aristas que salen de donde fue descubierto v.
- Este proceso continúa hasta que se descubran todos los vértices alcanzables desde el vértice original.
- Si aún existen vértices sin visitar el DFS selecciona alguno de ellos como nuevo origen y repite la búsqueda.

DFS

¿Cómo se forma el grafo de predecesores en el DFS?

DFS

¿Cómo se forma el grafo de predecesores en el DFS?

- Como en el BFS, siempre que DFS descubre un vértice v durante el escaneo de la lista de adyacencia de un vértice u que ya ha sido descubierto, registra este evento estableciendo el atributo predecesor de v $v.\pi = u$.

DFS

¿Cómo se forma el grafo de predecesores en el DFS?

- Como en el BFS, siempre que DFS descubre un vértice v durante el escaneo de la lista de adyacencia de un vértice u que ya ha sido descubierto, registra este evento estableciendo el atributo predecesor de v $v.\pi = u$.

¿Qué forma toma el grafo de predecesores producido por el DFS?

DFS

¿Cómo se forma el grafo de predecesores en el DFS?

- Como en el BFS, siempre que DFS descubre un vértice v durante el escaneo de la lista de adyacencia de un vértice u que ya ha sido descubierto, registra este evento estableciendo el atributo predecesor de v $v.\pi = u$.

¿Qué forma toma el grafo de predecesores producido por el DFS?

- A diferencia del BFS en el que el subgrafo de predecesores forma un árbol, el subgrafo de predecesores producido por DFS puede estar compuesto por varios árboles ya que la búsqueda puede repetirse con múltiples orígenes.

DFS

¿Cómo se define el subgrafo de predecesores producido por el DFS?

DFS

¿Cómo se define el subgrafo de predecesores producido por el DFS?

- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V, E_\pi)$ donde

DFS

¿Cómo se define el subgrafo de predecesores producido por el DFS?

- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V, E_\pi)$ donde

$$E_\pi = \{(v.\pi, v) \mid v \in V \wedge v.\pi \neq NIL\}$$

DFS

¿Cómo se define el subgrafo de predecesores producido por el DFS?

- Formalmente, para un grafo $G = (V, E)$ con origen s se define un subgrafo de predecesores G_π como $G_\pi = (V, E_\pi)$ donde

$$E_\pi = \{(v.\pi, v) \mid v \in V \land v.\pi \neq NIL\}$$

- El grafo de predecesores G_π producido por el DFS forma un bosque DF compuesto por varios árboles DF.

DFS

¿Cómo mantiene el DFS el registro de su progreso?

DFS

¿Cómo mantiene el DFS el registro de su progreso?

- Como en el BFS, el DFS colorea los vértices durante la búsqueda para indicar su estado.

DFS

¿Cómo mantiene el DFS el registro de su progreso?

- Como en el BFS, el DFS colorea los vértices durante la búsqueda para indicar su estado.
- Cada vértice es inicialmente blanco.

DFS

¿Cómo mantiene el DFS el registro de su progreso?

- Como en el BFS, el DFS colorea los vértices durante la búsqueda para indicar su estado.
- Cada vértice es inicialmente blanco.
- Cuando se descubre el vértice blanco se colorea de gris.

DFS

¿Cómo mantiene el DFS el registro de su progreso?

- Como en el BFS, el DFS colorea los vértices durante la búsqueda para indicar su estado.
- Cada vértice es inicialmente blanco.
- Cuando se descubre el vértice blanco se colorea de gris.
- El color del vértice gris cambia a negro cuando su lista de adyacencia ha sido examinada por completo.

DFS

¿Cómo mantiene el DFS el registro de su progreso?

- Como en el BFS, el DFS colorea los vértices durante la búsqueda para indicar su estado.
- Cada vértice es inicialmente blanco.
- Cuando se descubre el vértice blanco se colorea de gris.
- El color del vértice gris cambia a negro cuando su lista de adyacencia ha sido examinada por completo.
- Esta técnica garantiza que cada vértice termine exactamente en un árbol DF, de manera que estos árboles sean disjuntos.

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

- Le asigna timestamps a cada vértice.

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

- Le asigna timestamps a cada vértice.
- Cada vértice v tiene dos timestamps.

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

- Le asigna timestamps a cada vértice.
- Cada vértice v tiene dos timestamps.
  - La primera timestamp $v.d$ registra cuando v se ha descubierto por primera vez.

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

- Le asigna timestamps a cada vértice.
- Cada vértice v tiene dos timestamps.
  - La primera timestamp $v.d$ registra cuando v se ha descubierto por primera vez.
  - La segunda timestamp $v.f$ registra cuando la búsqueda finaliza de examinar la lista de adyacencia de v.

DFS

¿Qué produce el DFS adicionalmente al bosque DF?

- Le asigna timestamps a cada vértice.
- Cada vértice v tiene dos timestamps.
  - La primera timestamp $v.d$ registra cuando v se ha descubierto por primera vez.
  - La segunda timestamp $v.f$ registra cuando la búsqueda finaliza de examinar la lista de adyacencia de v.
  - Estas timestamps proveen información importante acerca de la estructura del grafo y generalmente sirven para razonar acerca del comportamiento del DFS.

DFS

¿Qué asumimos antes de presentar el DFS?

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo *u.color*.

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo $u.color$.
  - El predecesor en el atributo $u.\pi$

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo $u.color$.
  - El predecesor en el atributo $u.\pi$
  - Registra (con timestamps) cuando descubre u en el atributo $u.d$ y cuando termina el vértice u en $u.f$

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo $u.color$.
  - El predecesor en el atributo $u.\pi$
  - Registra (con timestamps) cuando descubre u en el atributo $u.d$ y cuando termina el vértice u en $u.f$
  - Los timestamps son enteros entre 1 y $2|V|$, ya que existe un evento de descubrimiento y uno de finalización para cada uno de los $|V|$ vértices.

DFS

¿Qué asumimos antes de presentar el DFS?

- El algoritmo que se presentará asume que el grafo de entrada $G = (V, E)$ se representa utilizando listas de adyacencia.
- Adiciona algunos atributos adicionales a cada vértice en el grafo.
  - Para cada vértice $u \in V$ el color se guarda en el atributo $u.color$.
  - El predecesor en el atributo $u.\pi$
  - Registra (con timestamps) cuando descubre u en el atributo $u.d$ y cuando termina el vértice u en $u.f$
  - Los timestamps son enteros entre 1 y $2|V|$, ya que existe un evento de descubrimiento y uno de finalización para cada uno de los $|V|$ vértices.
  - Para cada vértice $u \in V$ $u.d < u.f$.

DFS

- El vértice u es blanco antes del tiempo $u.d$, gris entre $u.d$ y $u.f$ y negro después.

DFS

- El vértice u es blanco antes del tiempo $u.d$, gris entre $u.d$ y $u.f$ y negro después.
- El grafo G puede ser dirigido o no dirigido.

DFS

- El vértice u es blanco antes del tiempo $u.d$, gris entre $u.d$ y $u.f$ y negro después.
- El grafo G puede ser dirigido o no dirigido.
- La variable tiempo es una variable global que se utiliza para poner las timestamps.

DFS

¿Cómo sería el pseudocódigo para el DFS?

DFS

¿Cómo sería el pseudocódigo para el DFS?

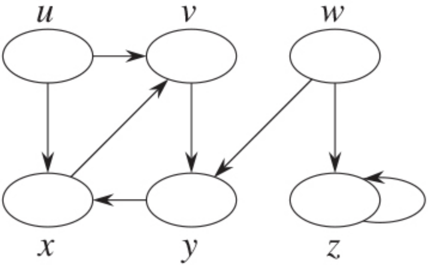
```
DFS(G)
for each vertex u ∈ G.V
    u.color = WHITE
    u.π = NIL
time = 0
for each vertex u ∈ G.V
    if u.color == WHITE
        DFS-VISIT(G, u)

DFS-VISIT(G, u)
time = time + 1              // white vertex u has just been discovered
u.d = time
u.color = GRAY
for each v ∈ G.Adj[u]        // explore edge (u, v)
    if v.color == WHITE
        v.π = u
        DFS-VISIT(G, v)
u.color = BLACK              // blacken u; it is finished
time = time + 1
u.f = time
```

DFS

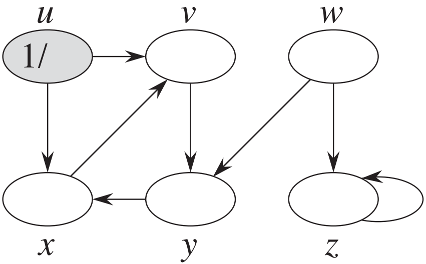
¿Cómo sería el funcionamiento del DFS en un grafo dirigido?

DFS

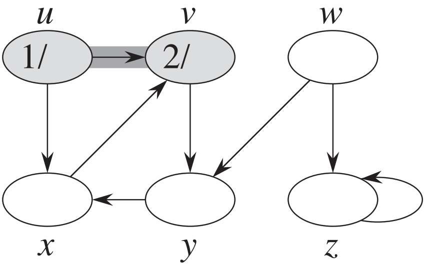
¿Cómo sería el funcionamiento del DFS en un grafo dirigido?



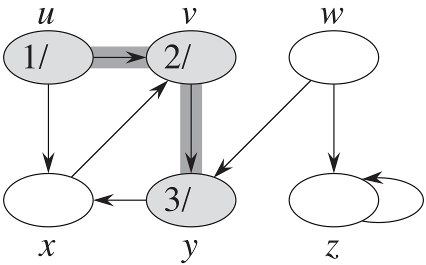
DFS



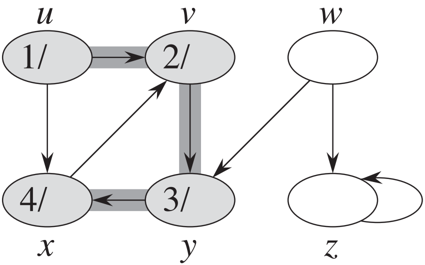
DFS



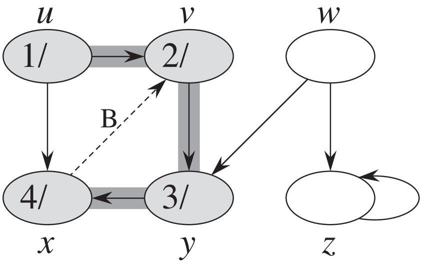
DFS



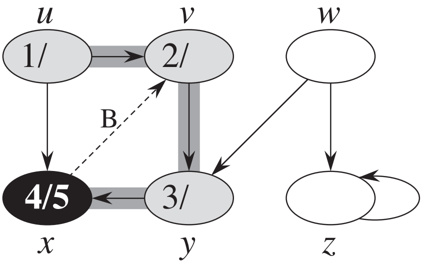
DFS



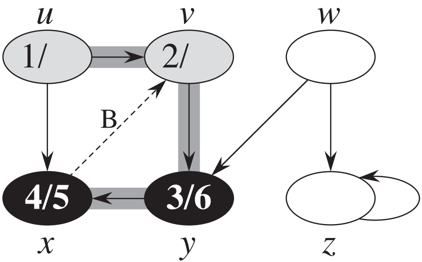
DFS



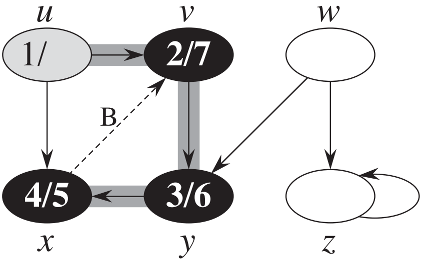
DFS



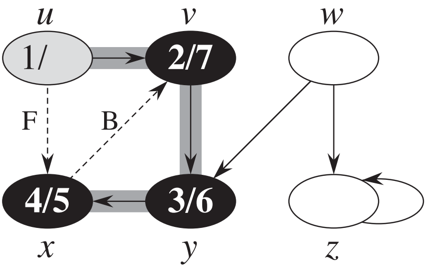
DFS



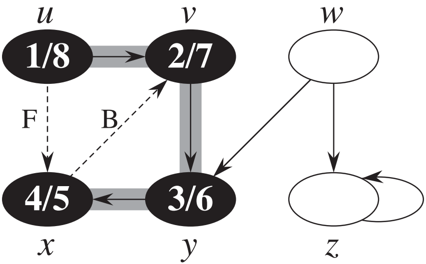
DFS



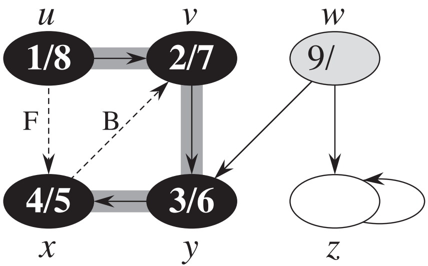
DFS



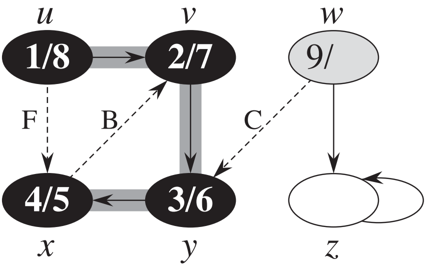
DFS



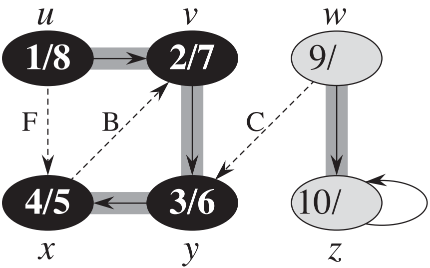
DFS



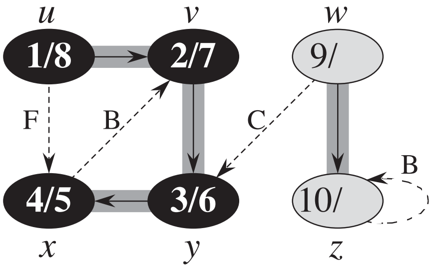
DFS



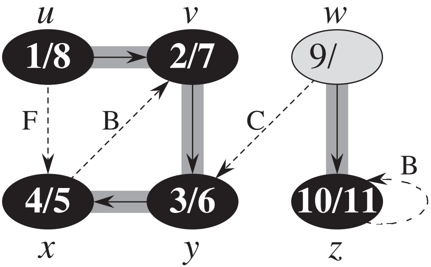
DFS



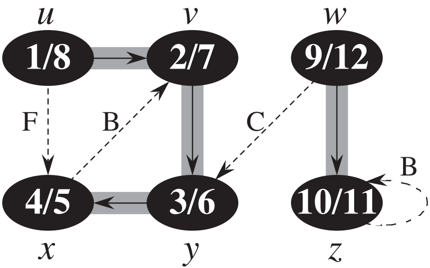
DFS



DFS



DFS



DFS

¿Cuál es la complejidad temporal del DFS?

DFS

¿Cuál es la complejidad temporal del DFS?

- $O(V + E)$.

DFS

¿Cuáles son algunas de las más importantes propiedades que se desprenden del DFS?

DFS

¿Cuáles son algunas de las más importantes propiedades que se desprenden del DFS?

- Tal vez la propiedad básica más importante del DFS es que el subgrafo de predecesores G_π forma un bosque de árboles.

DFS

¿Cuáles son algunas de las más importantes propiedades que se desprenden del DFS?

- Tal vez la propiedad básica más importante del DFS es que el subgrafo de predecesores G_π forma un bosque de árboles.
  - Ya que la estructura de los árboles DF refleja exactamente la estructura de las llamadas recursivas a DFS-VISIT.

DFS

¿Cuáles son algunas de las más importantes propiedades que se desprenden del DFS?

- Tal vez la propiedad básica más importante del DFS es que el subgrafo de predecesores G_π forma un bosque de árboles.
  - Ya que la estructura de los árboles DF refleja exactamente la estructura de las llamadas recursivas a DFS-VISIT.
  - Esto es que $u = v.\pi$ si sólo si DFS-VISIT(G,v) fue llamado durante la búsqueda sobre la lista de adyacencia de u.

DFS

¿Cuáles son algunas de las más importantes propiedades que se desprenden del DFS?

- Tal vez la propiedad básica más importante del DFS es que el subgrafo de predecesores G_π forma un bosque de árboles.
  - Ya que la estructura de los árboles DF refleja exactamente la estructura de las llamadas recursivas a DFS-VISIT.
  - Esto es que $u = v.\pi$ si sólo si DFS-VISIT(G,v) fue llamado durante la búsqueda sobre la lista de adyacencia de u.
  - Adicionalmente el vértice v es descendiente del vértice u en el bosque DF si y sólo si se descubre en el lapso en que u es gris.

DFS

- Otra propiedad importante del DFS es que los tiempos de descubrimiento y terminación tienen estructura de paréntesis.

DFS

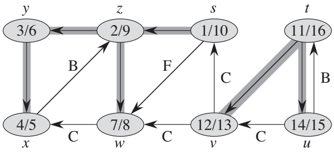
- Otra propiedad importante del DFS es que los tiempos de descubrimiento y terminación tienen estructura de paréntesis.
  - Si representamos el descubrimiento de un vértice u con un paréntesis (u y representamos su terminación con u), la historia de descubrimientos y terminaciones resultará en una expresión bien formada en cuanto el encadenamiento de paréntesis.

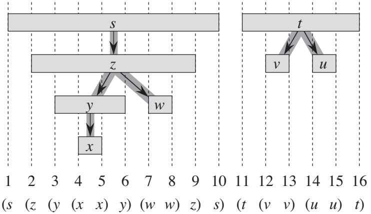
DFS

¿Cómo sería el resultado de los paréntesis al aplicar un DFS sobre un grafo dirigido?

DFS

¿Cómo sería el resultado de los paréntesis al aplicar un DFS sobre un grafo dirigido?





DFS

¿Qué otra propiedad se desprende del DFS?

DFS

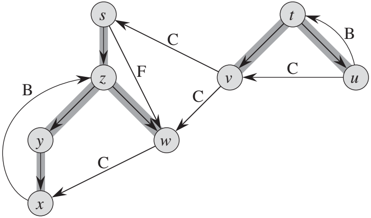
¿Qué otra propiedad se desprende del DFS?

- Clasificación de aristas

DFS

¿Qué otra propiedad se desprende del DFS?

- Clasificación de aristas
  - Aristas de descubrimiento: son aquellas aristas que conducen al descubrimiento de nuevos vértices. También se les llama aristas de árbol.
  - Aristas de retorno: son las aristas que conducen a vértices antecesores ya visitados en el árbol.
  - Aristas de avance: son las aristas que conducen a vértices descendientes en el árbol.
  - Aristas de cruce: son aristas que conducen a un vértice que no es ancestro de otro o a vértices que están en diferentes árboles.



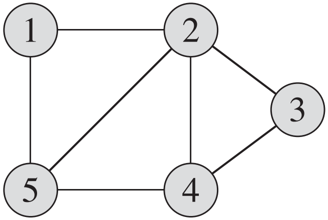
Agenda del día

1. **Búsquedas en grafos**
   - Introducción
   - BFS
   - Ejercicios
   - DFS
   - Ejercicios

Ejercicios

Ejercicio

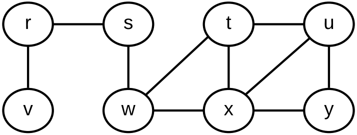
Muestre los valores d y π *que resultan de correr el BFS sobre el siguiente grafo no dirigido cuando el origen es el vértice* 3.



Ejercicios

Ejercicio

Muestre los valores d y π que resultan de correr el BFS sobre el siguiente grafo no dirigido cuando el origen es el vértice u.



Ejercicios

Ejercicio

Muestre como funciona el DFS en el siguiente grafo. Asuma que para el ciclo de las líneas 5-7 del DFS, se consideran los vértices en orden alfabético y asuma que cada lista de adyacencia está ordenada alfabéticamente. Muestre los tiempos de descubrimiento y finalización.