

# Algoritmo de búsqueda primero en anchura

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad  
Universidad Nacional Autónoma de México (UNAM)

## Algoritmo de búsqueda primero en anchura

2018-09-14

Algoritmo de búsqueda primero en anchura

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad  
Universidad Nacional Autónoma de México (UNAM)

El segundo de los algoritmos que presentaremos es el denominado algoritmo de búsqueda primero en anchura.

DFS y BFS son el basamento sobre el que podremos construir algoritmos más sofisticados.

# Algoritmo primero en anchura (Breadth-first search)

## BFS definido a partir del algoritmo DFS

La secuencia de operaciones que definen al algoritmo BFS es la misma que para el algoritmo DFS, la diferencia entre ambos algoritmos es que **BFS utiliza una cola para almacenar los estados de la agenda** a diferencia de DFS que usa una pila. La estructura de datos que almacena los nodos expandidos es la misma en ambos algoritmos.

## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

BFS definido a partir del algoritmo DFS

La secuencia de operaciones que definen al algoritmo BFS es la misma que para el algoritmo DFS, la diferencia entre ambos algoritmos es que BFS utiliza una cola para almacenar los estados de la agenda a diferencia de DFS que usa una pila. La estructura de datos que almacena los nodos expandidos es la misma en ambos algoritmos.

Para construir el algoritmo BFS vamos a partir del algoritmo DFS que ya ilustramos con un ejemplo anteriormente.

La secuencia de pasos a seguir es la misma en ambos algoritmos.

Solo tenemos que cambiar la estructura de datos que usamos para representar la agenda.

BFS utiliza una cola en lugar de una pila.

## Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

## Algoritmo de búsqueda primero en anchura

2018-09-14

└ Algoritmo primero en anchura (Breadth-first search)

Recordemos como funciona una cola.

# Algoritmo primero en anchura (Breadth-first search)

## Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que

## Algoritmo de búsqueda primero en anchura

2018-09-14

### └ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que

Al ingresar elementos a una cola, estos se agregan al final de la lista.

# Algoritmo primero en anchura (Breadth-first search)

## Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.

# Algoritmo de búsqueda primero en anchura

2018-09-14

## └ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.

Mientras que al sacarlos, se extrae el que esta al frente.

## Algoritmo primero en anchura (Breadth-first search)

### Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.

 ← cola vacía

## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.

 ← cola vacía

Como ejemplo consideremos una cola vacía, es decir, sin elementos.

# Algoritmo primero en anchura (Breadth-first search)

## Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



frente

 $\text{agregar}(B)$ 

## Algoritmo de búsqueda primero en anchura

2018-09-14

## Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



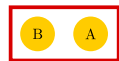
frente

 $\text{agregar}(B)$ Vamos a agregar el nodo  $B$

## Algoritmo primero en anchura (Breadth-first search)

### Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



frente

*agregar(A)*

## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



frente

*agregar(A)*

Agregamos  $A$  al final de la cola.



## Algoritmo primero en anchura (Breadth-first search)

### Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



*agregar( $G$ )*

## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Luego  $G$ .

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



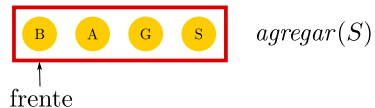
*agregar( $G$ )*

frente

## Algoritmo primero en anchura (Breadth-first search)

### Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Y el nodo  $S$ .

Vemos que el frente de la cola sigue apuntando al nodo ingresado primero.

Agenda es una cola

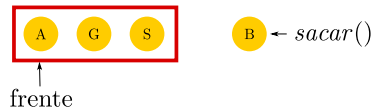
- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



## Algoritmo primero en anchura (Breadth-first search)

### Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



## Algoritmo de búsqueda primero en anchura

2018-09-14

### Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Agenda es una cola

- La operación agregar ingresa un nuevo estado al final de la lista de estados, mientras que
- la operación sacar extrae el elemento que esta al frente de la lista.



Al aplicar la operación *sacar* extraemos el elemento al frente.  
En este caso el nodo *B*.

El nuevo frente ahora es el segundo elemento ingresado.

2018-09-14<sup>o</sup>

- └ Algoritmo primero en anchura (Breadth-first search)

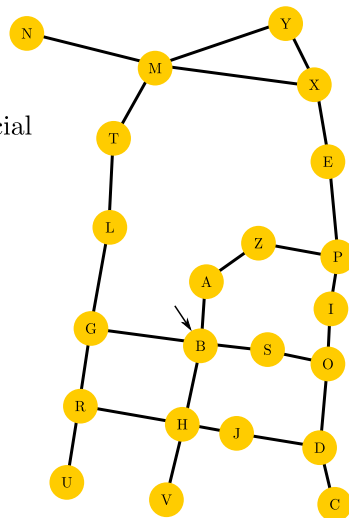
**Algoritmo primero en anchura (Breadth-first search)**

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda  
[B] ← agregar(B) Estado inicial

Examinados

Inicializamos la agenda con el estado inicial.



Expandidos

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

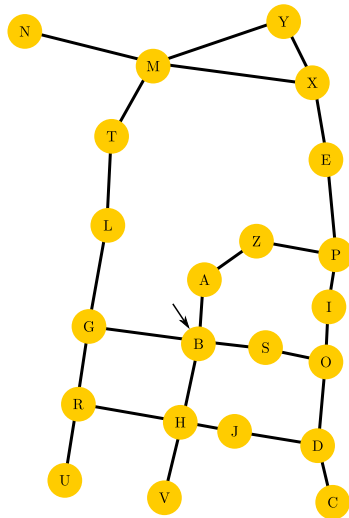
Agenda

[~~X~~]

*sacar()*

Expandidos

[B]



## Algoritmo de búsqueda primero en anchura

2018-09-14

└ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

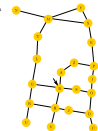
Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

sort()

Expandidos

[B]



Vamos a iterar hasta que se logre la meta o se acaben los estados almacenados en la agenda.

Sacamos el único elemento en la agenda.

Lo agregamos al conjunto de estados expandidos.

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

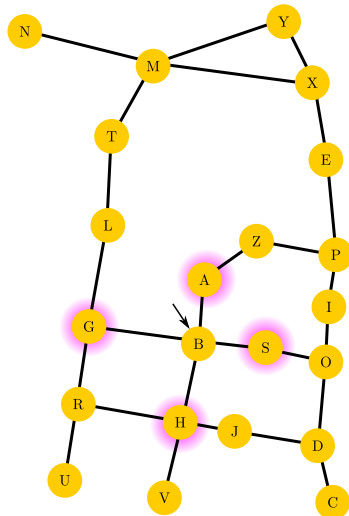
Agenda

$[X]$

$expandir(B)$

$[G_B \quad A_B \quad S_B \quad H_B]$

Expandidos  
 $[B]$



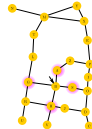
## Algoritmo de búsqueda primero en anchura

2018-09-14

└ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)  
Agenda  
 $[X]$   $expandir(B)$   
 $[G_B \quad A_B \quad S_B \quad H_B]$



Expandidos  
 $[B]$

Expandimos  $B$ , obteniendo 4 sucesores.  
Verificamos si alguno es el objetivo.  
No es el caso.  
Verificamos si no se han expandido.  
No es el caso.  
Los agregamos todos a la agenda.

## Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

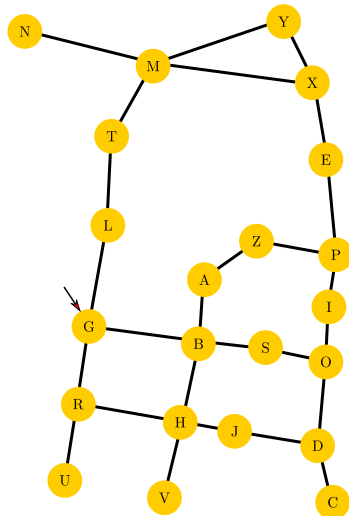
Agenda

$[X]$  *sacar()*

$[X_B \quad A_B \quad S_B \quad H_B]$

↑  
frente

Expandidos  
 $[B \quad G]$



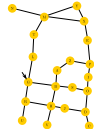
## Algoritmo de búsqueda primero en anchura

2018-09-14

└ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)  
Agenda  
 $[X]$  *sacar()*  
 $[X_B \quad A_B \quad S_B \quad H_B]$   
↑  
frente



Expandidos  
 $[B \quad G]$

Empezamos la siguiente iteración.

Sacamos el elemento al frente de la cola.

Se obtiene el nodo  $G$ .

Se agrega al conjunto de expandidos.

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

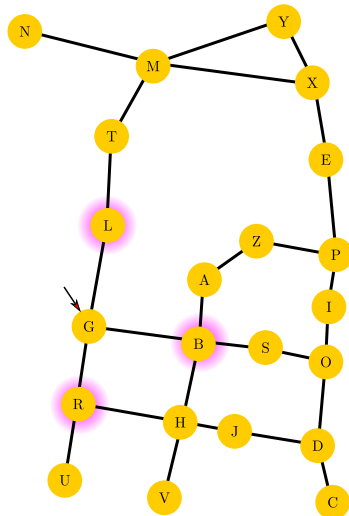
Agenda

$\boxed{\emptyset}$   $expandir(G)$

$\boxed{G_B \quad A_B \quad S_B \quad H_B}$

$[A_B \quad S_B \quad R_H \quad L_G \quad R_G]$

Expandidos  
 $[B \quad G]$



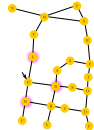
## Algoritmo de búsqueda primero en anchura

2018-09-14

└ Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)  
Agenda  
 $\boxed{\emptyset}$   $expandir(G)$   
 $\boxed{G_B \quad A_B \quad S_B \quad H_B}$   
 $[A_B \quad S_B \quad R_H \quad L_G \quad R_G]$



Expandidos  
 $[B \quad G]$

Expandimos el nodo  $G$ .

Tiene dos sucesores no expandidos previamente  $L$  y  $R$ .

Ninguno es el objetivo.

Los agregamos a la agenda.



2018-09-14<sup>o</sup>

- └ Algoritmo primero en anchura (Breadth-first search)

**Algoritmo primero en anchura (Breadth-first search)**

Algoritmo primero en anchura  
(Breadth-first search - BFS)

```

Agenda      start()
[ ] AB SB HB
[ ] SB HB LB RB
    
```

Expandido  
[ B G A ]

Nueva iteración.

Sacamos el elemento al frente de la cola, en este caso  $A$ .

Se agrega a expandidos.

$$\begin{bmatrix} B & G & A \end{bmatrix}$$

## Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[A]~~ *expandir(A)*

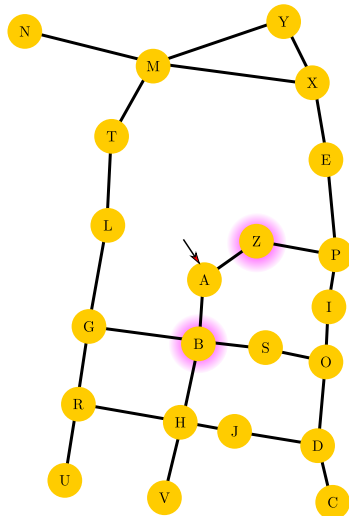
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~

~~[A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~

[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]

Expandidos

[B G A]



## Algoritmo de búsqueda primero en anchura

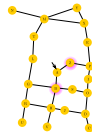
2018-09-14

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)  
Agenda  
~~[A]~~ *expandir(A)*  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~  
~~[A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~  
[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]

Expandidos  
[B G A]



Expandimos A.

Tiene dos sucesores.

B no se agrega pues ya se expandió.

Agregamos a Z al final de la cola.

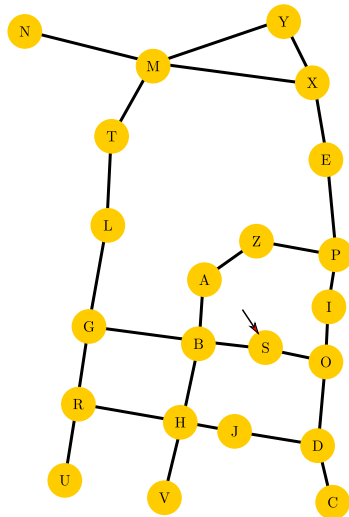
2018-09-14<sup>o</sup>

- └ Algoritmo primero en anchura (Breadth-first search)

Nueva iteración.  
Sacamos a  $S$  del frente de la cola.  
Lo agregamos a expandidos.

$$\begin{bmatrix} \mathbb{Q}_B & A_B & S_B & H_B \end{bmatrix}$$
$$\begin{bmatrix} \cancel{A}_B & S_B & H_B & L_G & R_G \end{bmatrix}$$
$$\begin{bmatrix} \cancel{S}_B & H_B & L_G & R_G & Z_A \end{bmatrix}$$

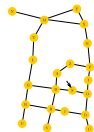
A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

$$\begin{bmatrix} B & G & A & S \end{bmatrix}$$


Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

<del>X</del>				sucro()
<del>X</del> <sub>0</sub>	A <sub>B</sub>	S <sub>B</sub>	H <sub>B</sub>	
<del>X</del> <sub>1</sub>	S <sub>D</sub>	H <sub>D</sub>	L <sub>G</sub>	R <sub>G</sub>
<del>X</del> <sub>6</sub>	H <sub>B</sub>	L <sub>G</sub>	R <sub>G</sub>	Z <sub>A</sub>



# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[S]~~ *expandir(S)*

~~[S<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~

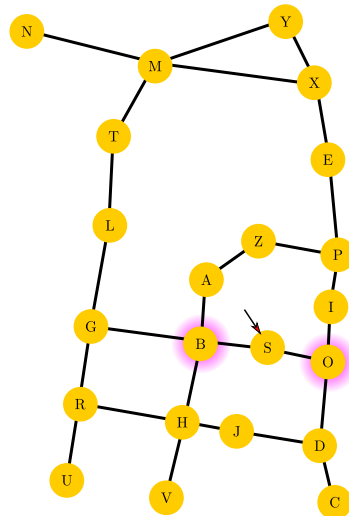
~~[S<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~

~~[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~

[H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]

Expandidos

[B G A S]



## Algoritmo de búsqueda primero en anchura

2018-09-14

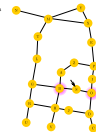
Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda  
~~[S]~~ *expandir(S)*  
~~[S<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~  
~~[S<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~  
~~[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~  
[H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]

Expandidos  
[B G A S]



Vamos a expandir *S*.  
Tiene dos sucesores *B* y *O*.  
Ninguno es el objetivo.  
*B* ya se expandió.  
Agregamos a *O*.

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[N]~~ *sacar()*

~~[G<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~

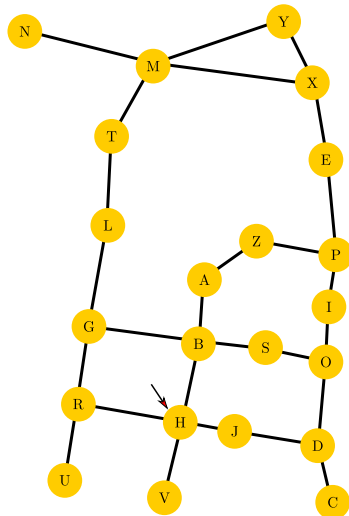
~~[A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~

~~[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~

~~[H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~

Expandidos

[B G A S H]



## Algoritmo de búsqueda primero en anchura

2018-09-14

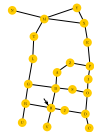
Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda  
~~[N]~~ *sacar()*  
~~[G<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~  
~~[A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~  
~~[S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~  
~~[H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~

Expandidos  
[B G A S H]



Nueva iteración.

Sacamos a *H* que está al frente de la cola.

Lo agregamos a expandidos.

## Algoritmo primero en anchura (Breadth-first search)

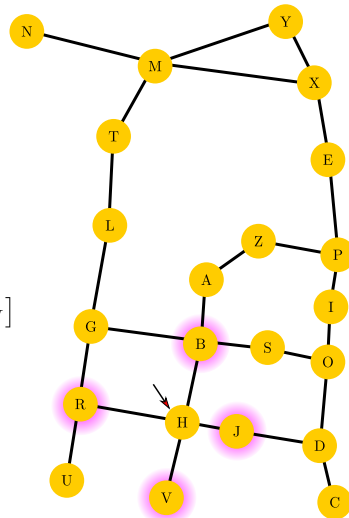
Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[ $\emptyset$ ]~~ *expandir(H)*  
~~[ $\emptyset_B$   $A_B$   $S_B$   $H_B$ ]~~  
~~[ $\emptyset_B$   $S_B$   $H_B$   $L_G$   $R_G$ ]~~  
~~[ $\emptyset_B$   $H_B$   $L_G$   $R_G$   $Z_A$ ]~~  
~~[ $\emptyset_B$   $L_G$   $R_G$   $Z_A$   $O_S$ ]~~  
 $[L_G \ R_G \ Z_A \ O_S \ R_H \ J_H \ V_H]$

Expandidos

$[B \ G \ A \ S \ H]$



## Algoritmo de búsqueda primero en anchura

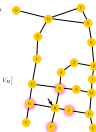
2018-09-14

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)  
 Agenda  
~~[ $\emptyset$ ]~~ *expandir(H)*  
~~[ $\emptyset_B$   $A_B$   $S_B$   $H_B$ ]~~  
~~[ $\emptyset_B$   $S_B$   $H_B$   $L_G$   $R_G$ ]~~  
~~[ $\emptyset_B$   $H_B$   $L_G$   $R_G$   $Z_A$ ]~~  
~~[ $\emptyset_B$   $L_G$   $R_G$   $Z_A$   $O_S$ ]~~  
 $[L_G \ R_G \ Z_A \ O_S \ R_H \ J_H \ V_H]$

Expandidos  
 $[B \ G \ A \ S \ H]$



Expandimos a  $H$ .

Tiene 4 sucesores.

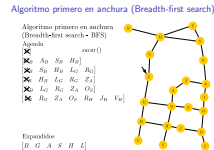
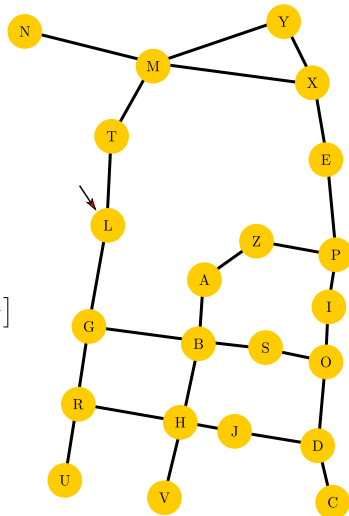
De estos,  $B$  ya se expandió.

Ninguno de los tres restantes es el objetivo.

Agregamos a  $R$ ,  $J$  y  $V$  a la agenda.

- └ Algoritmo primero en anchura (Breadth-first search)

Nueva iteración.  
Sacamos a  $L$  que esta al frente de la cola.  
Lo metemos a expandidos.



# Algoritmo primero en anchura (Breadth-first search)

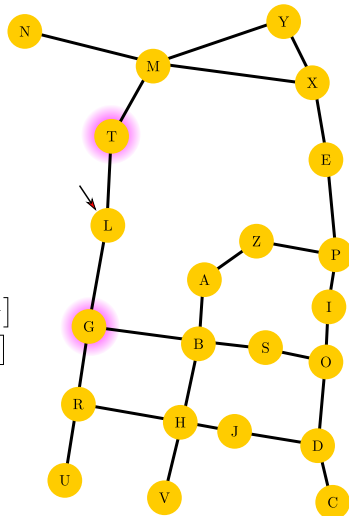
Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[X]~~ *expandir(L)*  
~~[X<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~  
~~[X<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~  
~~[X<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~  
~~[X<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~  
~~[L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub>]~~  
 [R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub>]

Expandidos

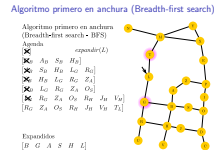
[B G A S H L]



## Algoritmo de búsqueda primero en anchura

2018-09-14

Algoritmo primero en anchura (Breadth-first search)



Vamos a expandir a *L*.

Tiene 2 sucesores.

Ninguno el objetivo.

Solo agregamos a *T*, *G* ya se había expandido.



2018-09-14<sup>o</sup>

- └ Algoritmo primero en anchura (Breadth-first search)

**Algoritmo primero en anchura (Breadth-first search)**

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

	hazcar()
✖	$A_D \ S_D \ H_D$
✖	$S_B \ H_B \ L_D \ R_D$
✖	$H_L \ L_C \ R_C \ Z_A$
✖	$L_D \ R_D \ Z_A \ O_S$
✖	$R_C \ Z_A \ O_S \ H_A$
✖	$Z_A \ O_S \ R_H \ V_H \ T_A$

Expandido

[ B G A S H L R ]

Nueva iteración.  
Sacamos a  $R$  que está al frente de la cola.  
Lo agregamos a expandidos.

$$[\cancel{A}_B \quad S_B \quad H_B \quad L_G \quad R_G]$$
$$\begin{bmatrix} \cancel{S}_B & H_B & L_G & R_G & Z_A \end{bmatrix}$$
$$\begin{bmatrix} \cancel{K}_B & L_G & R_G & Z_A & O_S \end{bmatrix}$$
$$[\cancel{L}_G \quad R_G \quad Z_A \quad O_S \quad R_H \quad J_H \quad V_H]$$
$$\begin{bmatrix} R_G & Z_A & O_S & R_H & J_H & V_H & T_L \end{bmatrix}$$

A graph with 20 nodes labeled A through T. The nodes are connected by edges forming a complex network. Node R is highlighted with a red arrow pointing to it.

$$[B \quad G \quad A \quad S \quad H \quad L \quad R]$$

2018-09-14<sup>c</sup>

- └ Algoritmo primero en anchura (Breadth-first search)

**Algoritmo primero en anchura (Breadth-first search)**

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Inicio expandir(R)

$H_0$	$A_0$	$S_0$	$H_0$
$S_0$	$H_0$	$L_0$	$R_0$
$H_0$	$L_0$	$R_0$	$Z_A$
$L_0$	$R_0$	$Z_A$	$O_0$
$R_0$	$Z_A$	$O_0$	$H_1$
$Z_A$	$O_0$	$H_1$	$V_1$
$O_0$	$H_1$	$V_1$	$T_1$
$H_1$	$V_1$	$T_1$	$U_1$

Expandir: [ B G A S H L R ]

Expandimos  $R$ .  
Solo agregamos a  $U$ .

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

$$\begin{bmatrix} \mathbb{X}_B & H_B & L_G & R_G & Z_A \end{bmatrix}$$
$$\begin{bmatrix} \cancel{X}_B & L_G & R_G & Z_A & O_S \end{bmatrix}$$
$$[\cancel{L}_G \quad R_G \quad Z_A \quad O_S \quad R_H \quad J_H \quad V_H]$$
$$\left[ \cancel{R_G} \quad Z_A \quad O_S \quad R_H \quad J_H \quad V_H \quad T_L \right]$$
$$\begin{bmatrix} Z_A & O_S & R_H & J_H & V_H & T_L & U_R \end{bmatrix}$$

Expandidos

$$[B \quad G \quad A \quad S \quad H \quad L \quad R]$$

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

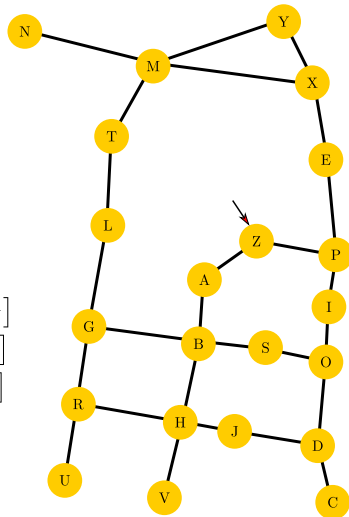
Agenda

~~[X]~~ *sacar()*

<del>[X]</del>	<del><math>A_B</math></del>	$S_B$	$H_B$			
<del><math>A_B</math></del>	$S_B$	$H_B$	$L_G$	$R_G$		
<del><math>S_B</math></del>	$H_B$	$L_G$	$R_G$	$Z_A$		
<del><math>H_B</math></del>	$L_G$	$R_G$	$Z_A$	$O_S$		
<del><math>L_G</math></del>	$R_G$	$Z_A$	$O_S$	$R_H$	$J_H$	$V_H$
<del><math>R_G</math></del>	$Z_A$	$O_S$	$R_H$	$J_H$	$V_H$	$T_L$
<del><math>Z_A</math></del>	$O_S$	$R_H$	$J_H$	$V_H$	$T_L$	$U_R$

Expandidos

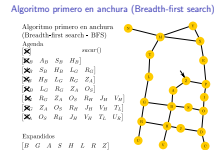
[B G A S H L R Z]



## Algoritmo de búsqueda primero en anchura

2018-09-14

Algoritmo primero en anchura (Breadth-first search)



Nueva iteración.

Sacamos a Z que esta al frente de la cola.

Lo agregamos a expandidos.

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[Z]~~ *expandir(Z)*

~~[~~Z~~<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~

~~[~~A~~<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~

~~[~~S~~<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~

~~[~~H~~<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~

~~[~~L~~<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub>]~~

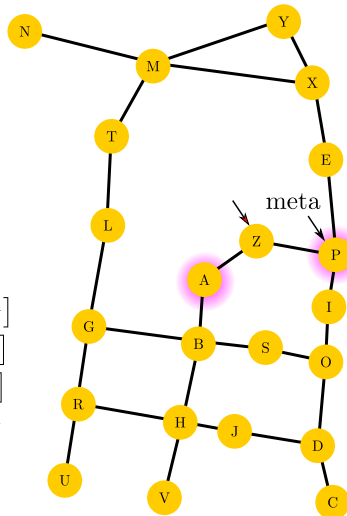
~~[~~R~~<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub>]~~

~~[~~Z~~<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>]~~

[O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>] → P<sub>Z</sub>

Expandidos

[B G A S H L R Z]



## Algoritmo de búsqueda primero en anchura

2018-09-14

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

Agenda

~~[Z]~~ *expandir(Z)*

~~[~~Z~~<sub>B</sub> A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~

~~[~~S~~<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~

~~[~~H~~<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~

~~[~~L~~<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~

~~[~~L~~<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub>]~~

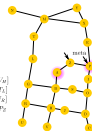
~~[~~Z~~<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub>]~~

~~[~~O~~<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>]~~

[O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>] → P<sub>Z</sub>

Expandidos

[B G A S H L R Z]



Expandimos a *Z*.

Vemos que *P*, su sucesor, es el objetivo.

Ya no lo agregamos a la cola.

Solo queda recuperar la ruta.

# Algoritmo primero en anchura (Breadth-first search)

Algoritmo primero en anchura  
(Breadth-first search - BFS)

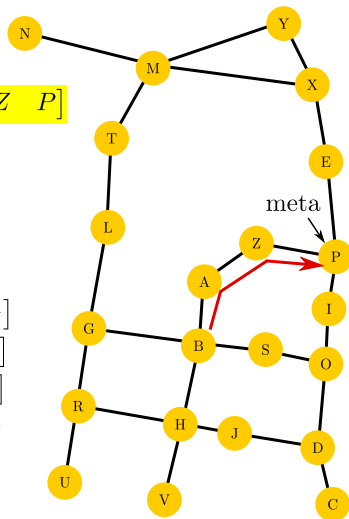
Agenda

ruta = [B A Z P]

~~[B]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub>]~~  
~~[B A<sub>B</sub> S<sub>B</sub> H<sub>B</sub> L<sub>G</sub> R<sub>G</sub> Z<sub>A</sub> O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>]~~  
 [O<sub>S</sub> R<sub>H</sub> J<sub>H</sub> V<sub>H</sub> T<sub>L</sub> U<sub>R</sub>] → P<sub>Z</sub>

Expandidos

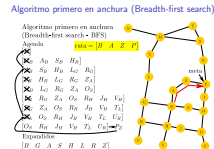
[B G A S H L R Z]



## Algoritmo de búsqueda primero en anchura

2018-09-14

Algoritmo primero en anchura (Breadth-first search)



Partiendo del objetivo  $P$ ,  
recuperamos la ruta recorriendo los antecesores de cada nodo.  
 $P$  proviene de  $Z$ .  
 $Z$  proviene de  $A$ .  
 $A$  proviene de  $B$ , el nodo inicial.  
La ruta final es  $B, A, Z, P$ .  
El algoritmo regresa esta ruta y termina.