

# CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ VÀ ỨNG DỤNG

- Tìm kiếm theo chiều sâu
- Tìm kiếm theo chiều rộng
- Một số ứng dụng

# TÌM KIẾM THEO CHIỀU SÂU (Depth-First Search-DFS)

- Thuật toán DFS
- Phân tích DFS

# THUẬT TOÁN DFS

Ý tưởng thuật toán

- Bắt đầu tìm kiếm từ **một đỉnh  $u$**  nào đó
- Chọn **đỉnh kề  $v$  tùy ý của  $u$**  để tiếp tục quá trình tìm kiếm và lặp lại quá trình tìm kiếm này đối với  $v$

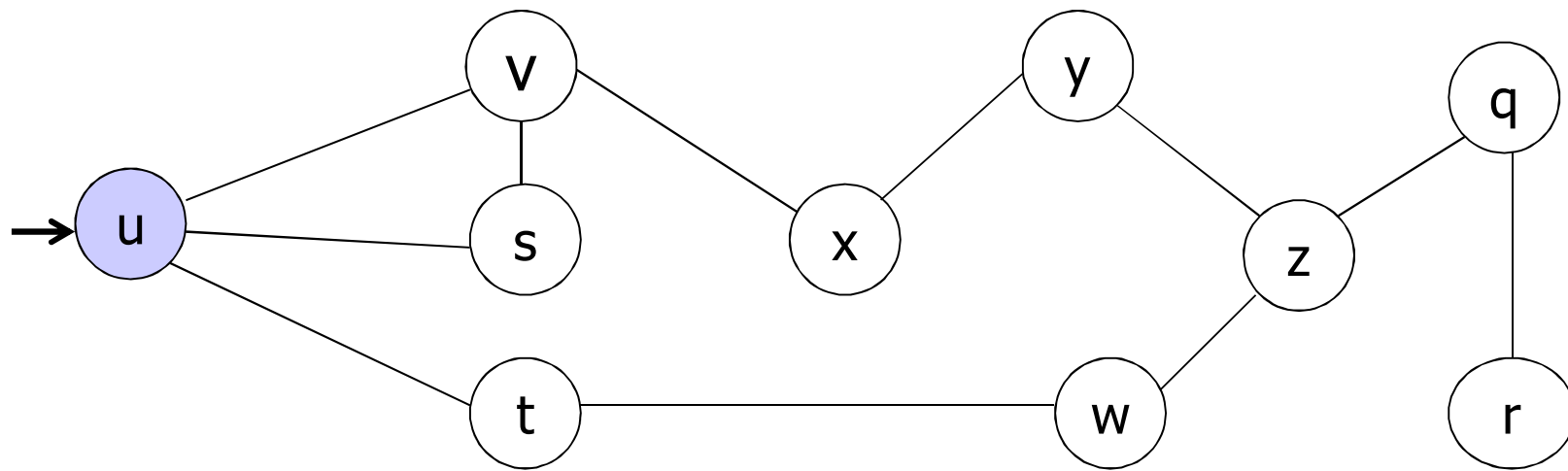
# THUẬT TOÁN DFS

Ý tưởng thuật toán

- Dùng các màu để không lặp lại các đỉnh tìm kiếm
- Dùng một mảng để lưu trữ đỉnh đi trước của đỉnh được tìm kiếm

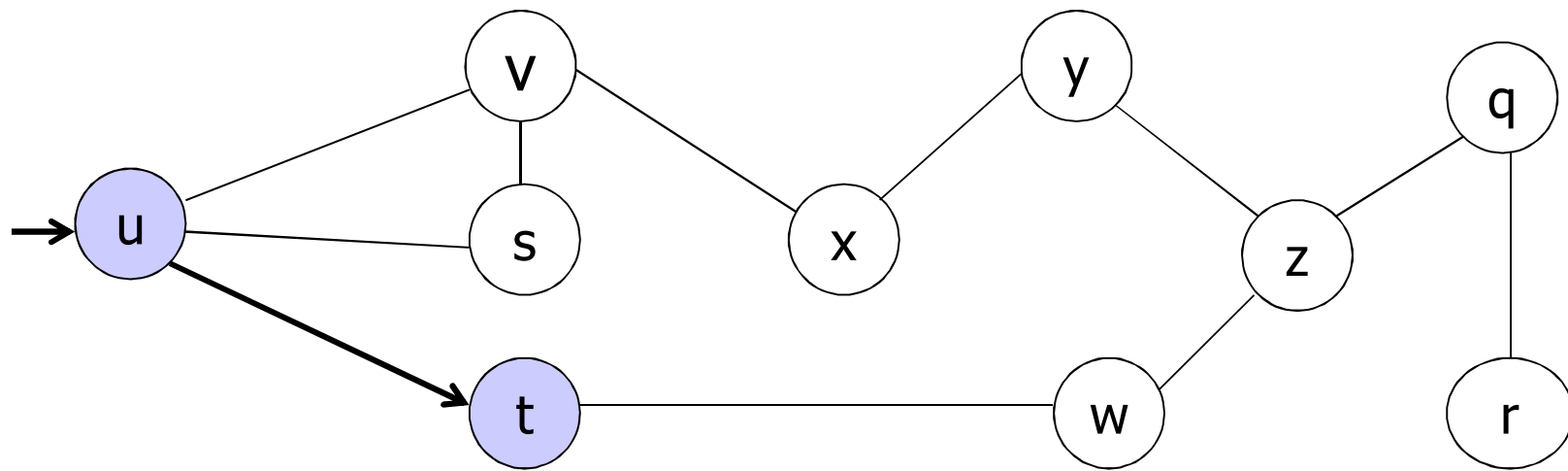
# THUẬT TOÁN DFS

Bắt đầu tìm kiếm từ đỉnh u (tô xám)



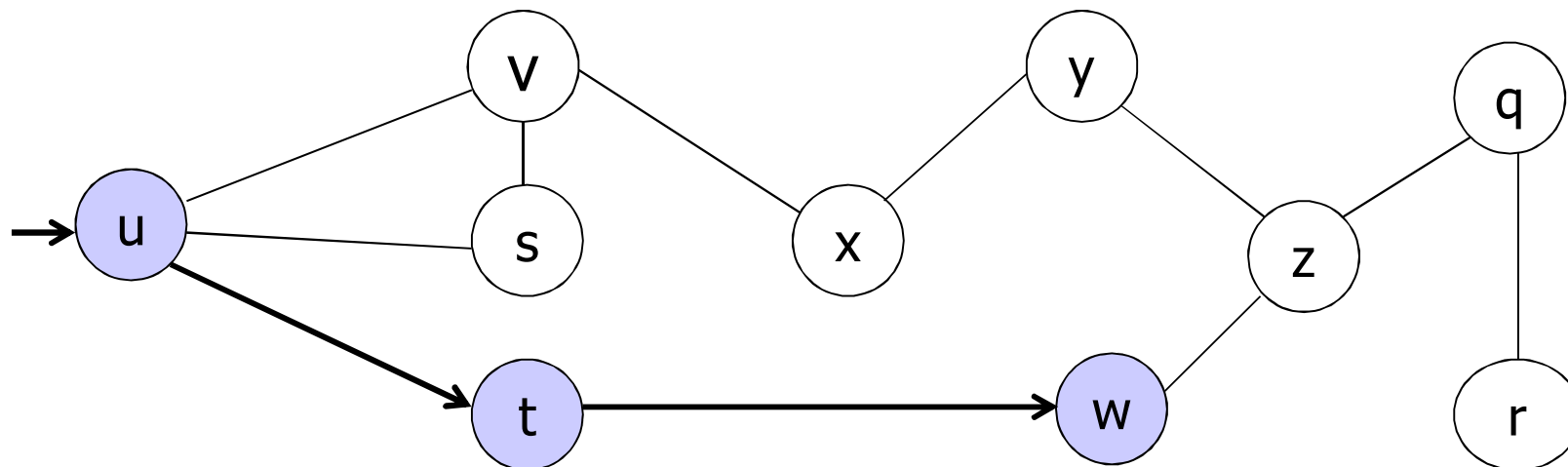
# THUẬT TOÁN DFS

Tìm thấy đỉnh t



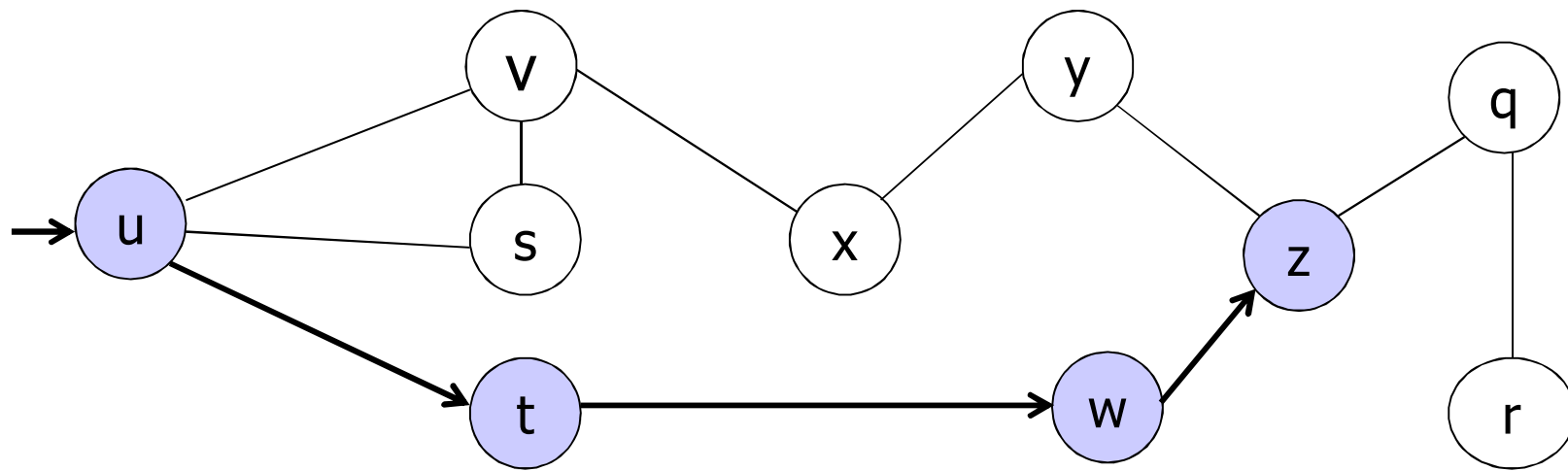
# THUẬT TOÁN DFS

Tìm thấy đỉnh w



# THUẬT TOÁN DFS

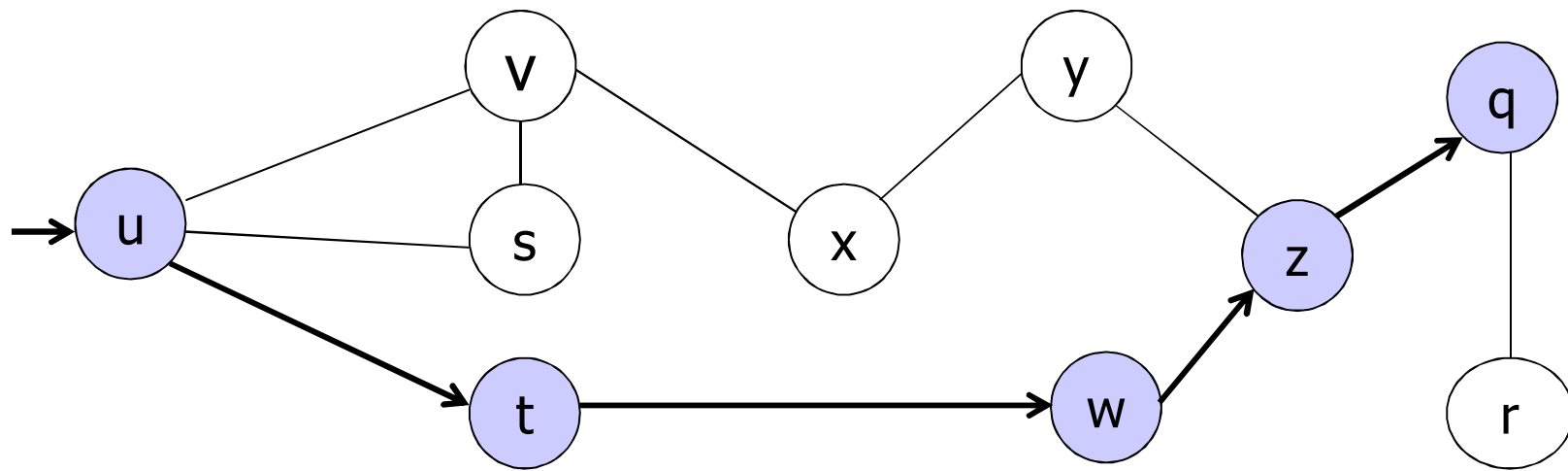
Tìm thấy đỉnh z





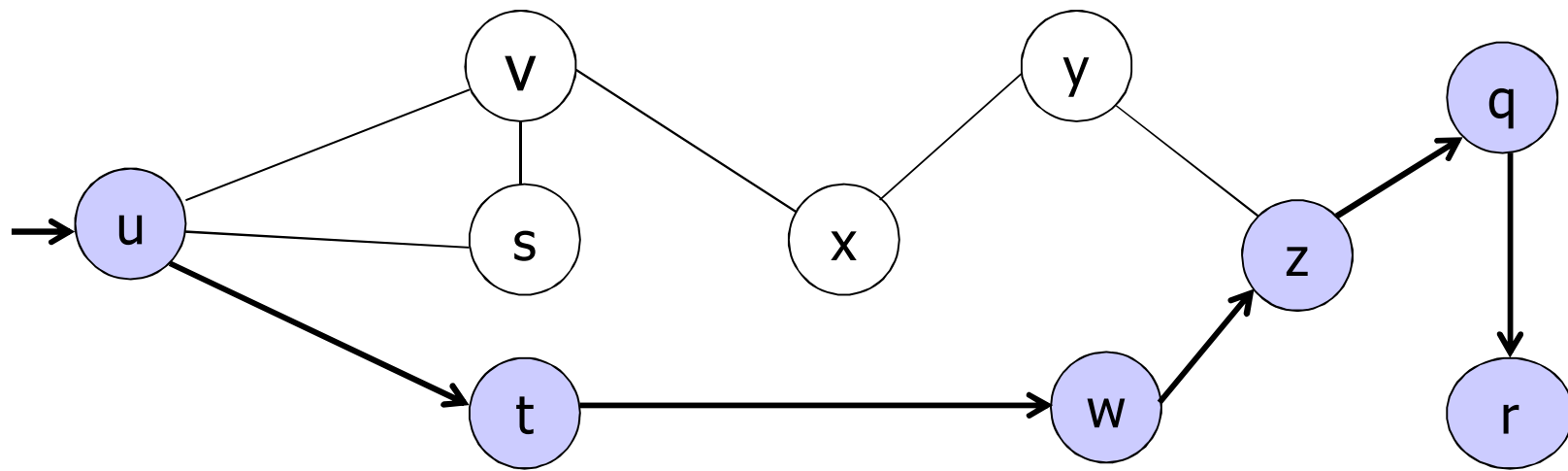
# THUẬT TOÁN DFS

Tìm thấy đỉnh q



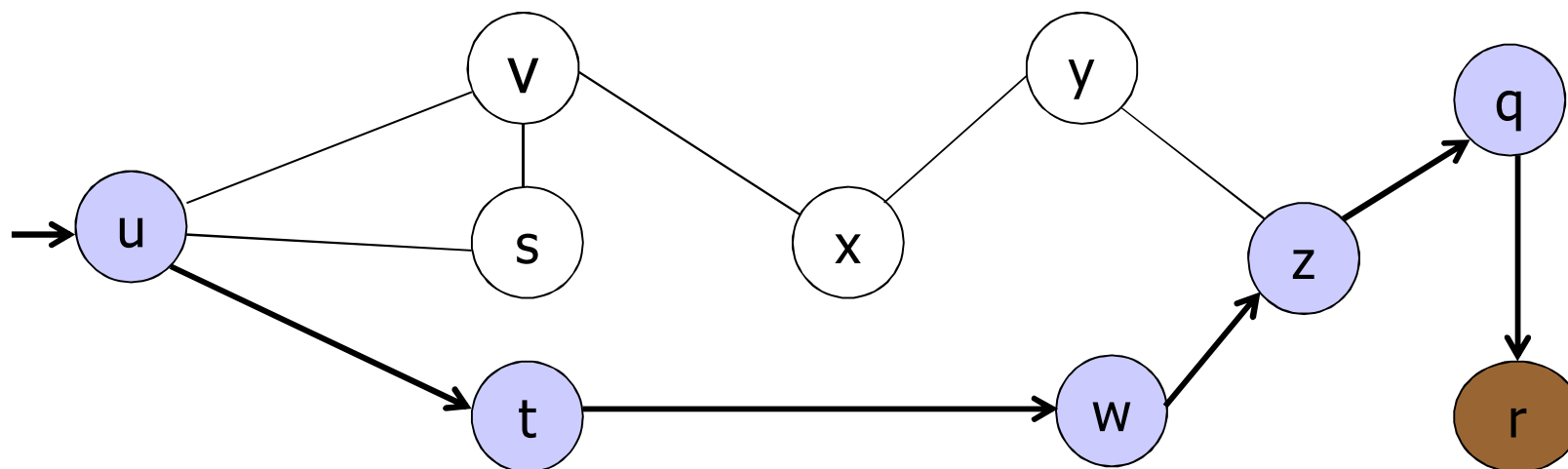
# THUẬT TOÁN DFS

Tìm thấy đỉnh r



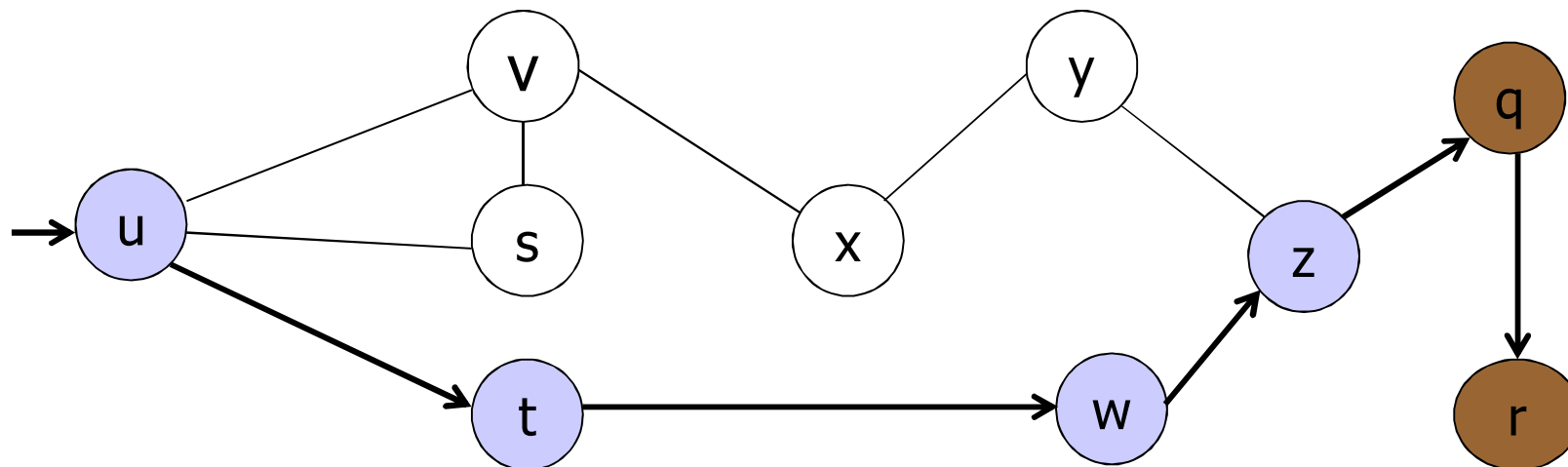
# THUẬT TOÁN DFS

Kết thúc (hoàn thành) tìm kiếm đỉnh r (tô nâu)



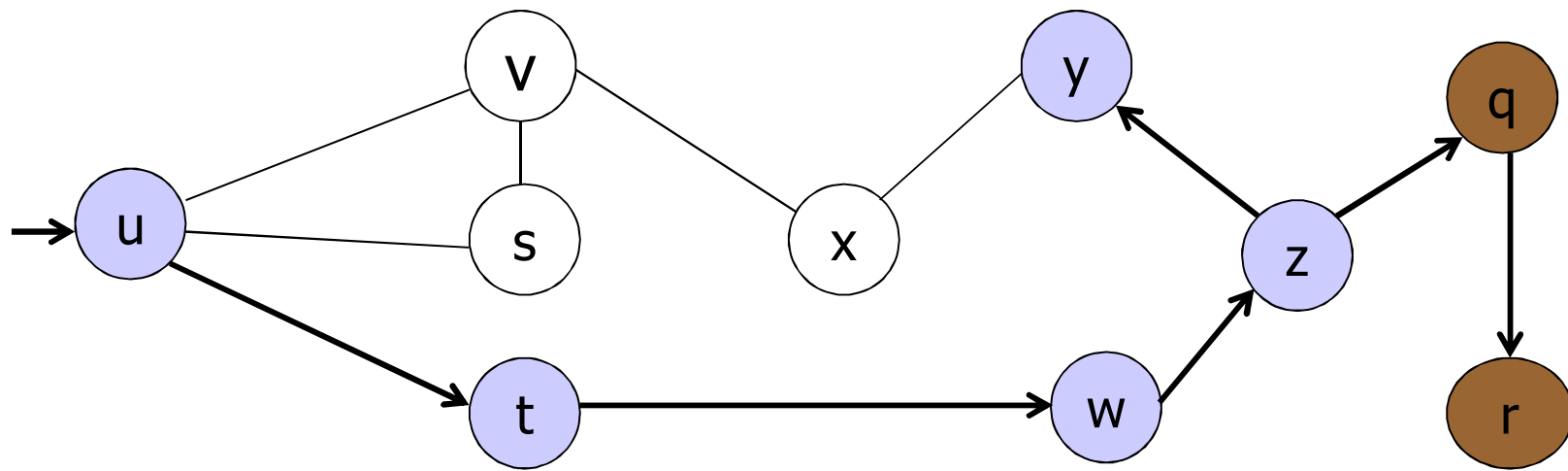
# THUẬT TOÁN DFS

Kết thúc (hoàn thành) tìm kiếm đỉnh q



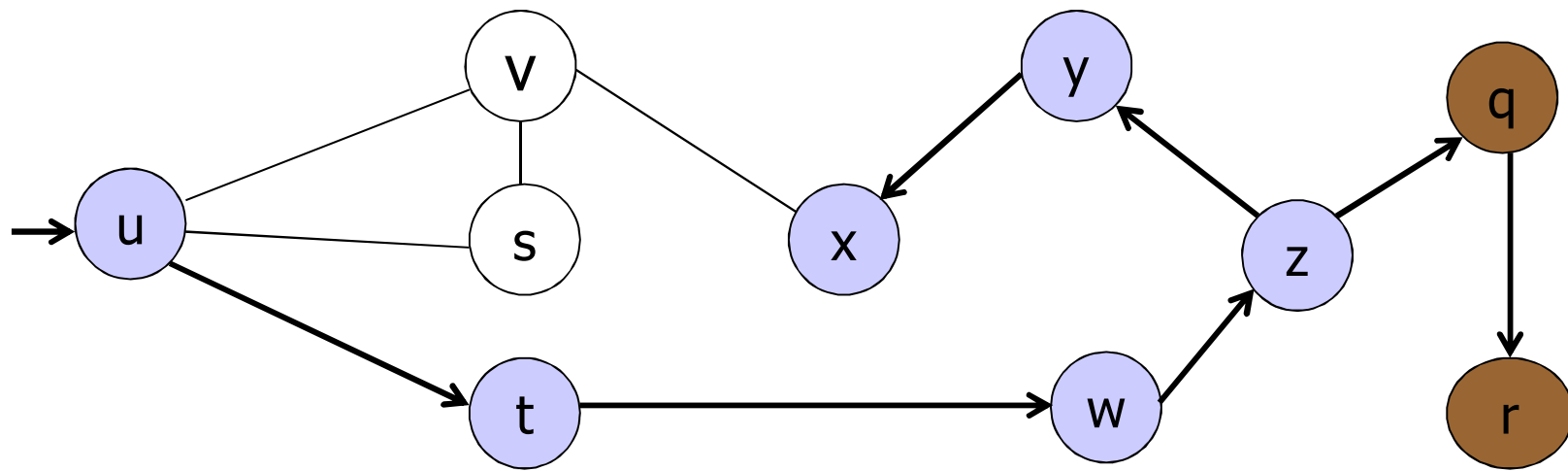
# THUẬT TOÁN DFS

Tìm thấy đỉnh y



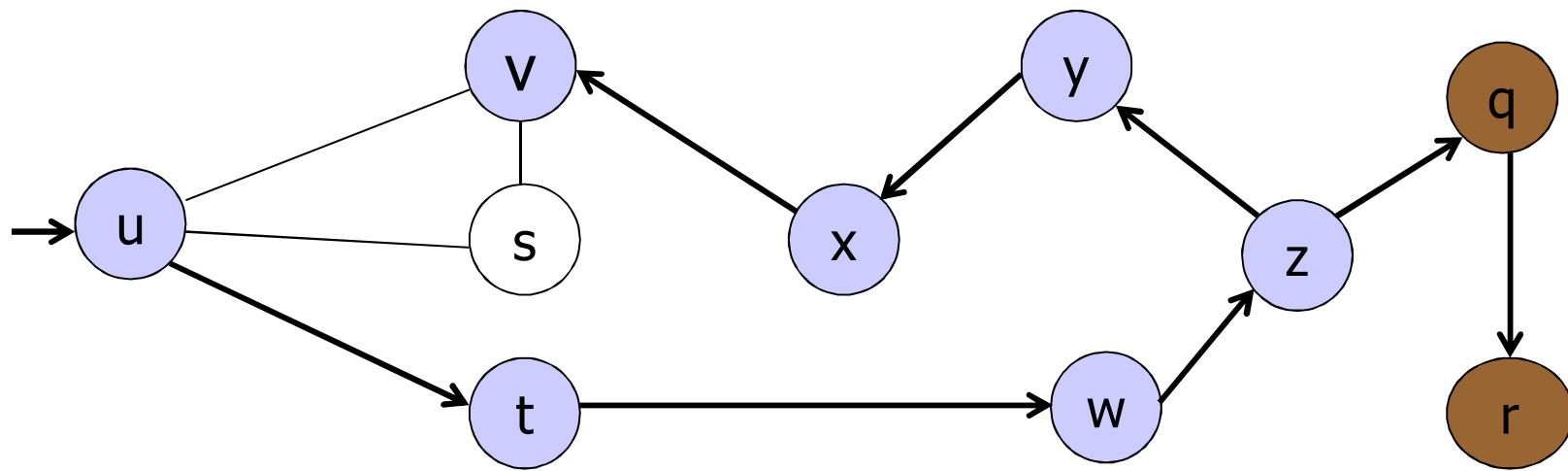
# THUẬT TOÁN DFS

Tìm thấy đỉnh x



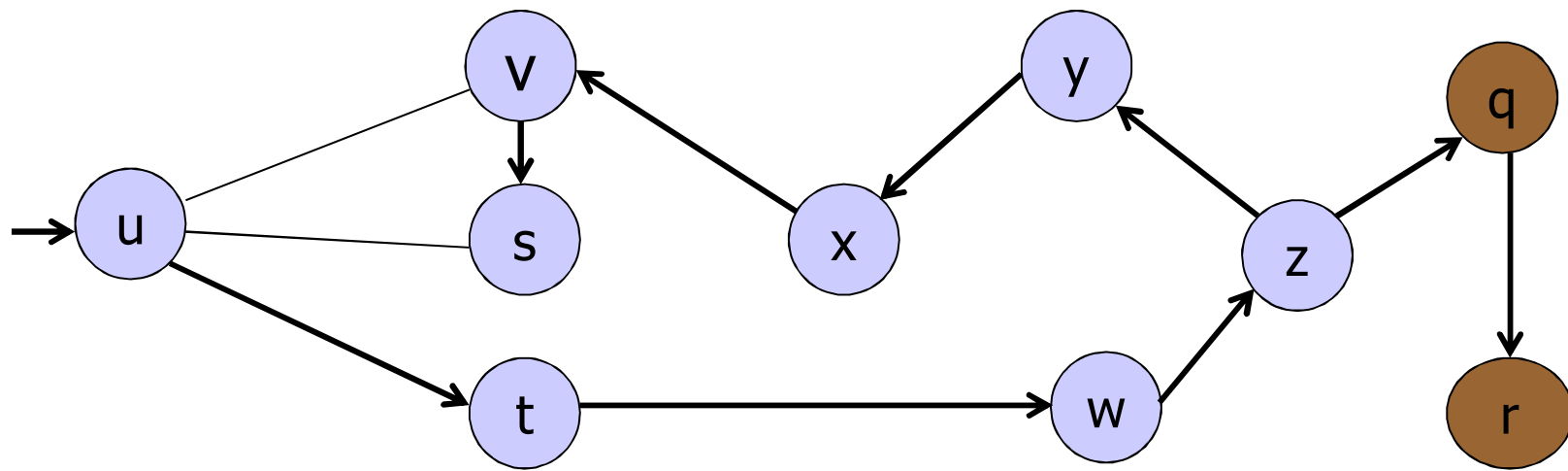
# THUẬT TOÁN DFS

Tìm thấy đỉnh v



# THUẬT TOÁN DFS

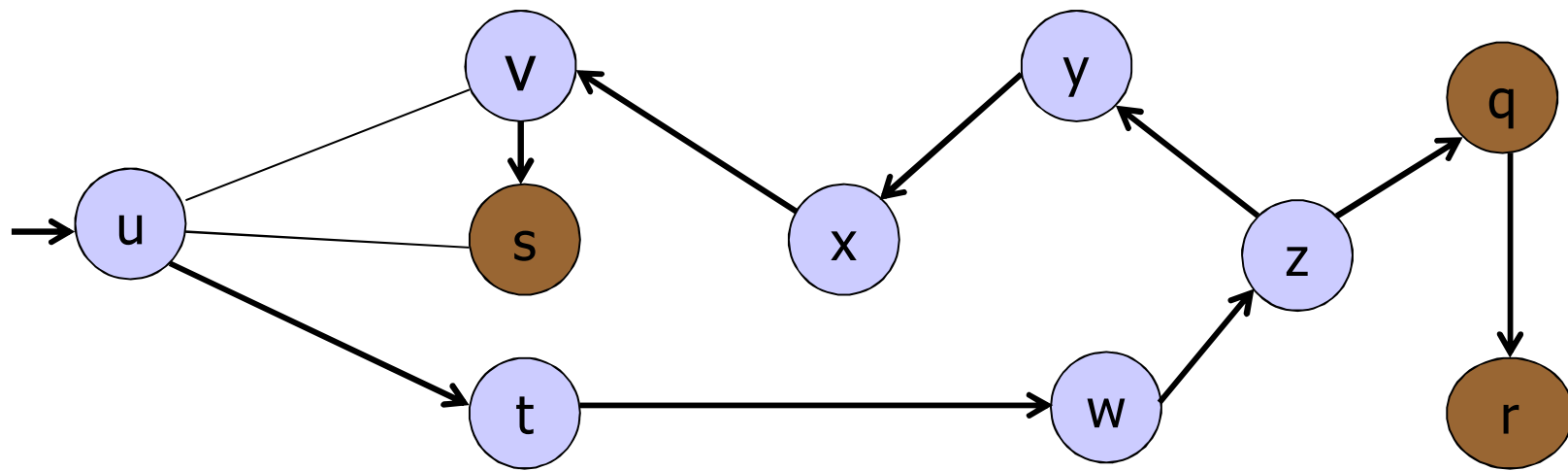
Tìm thấy đỉnh s





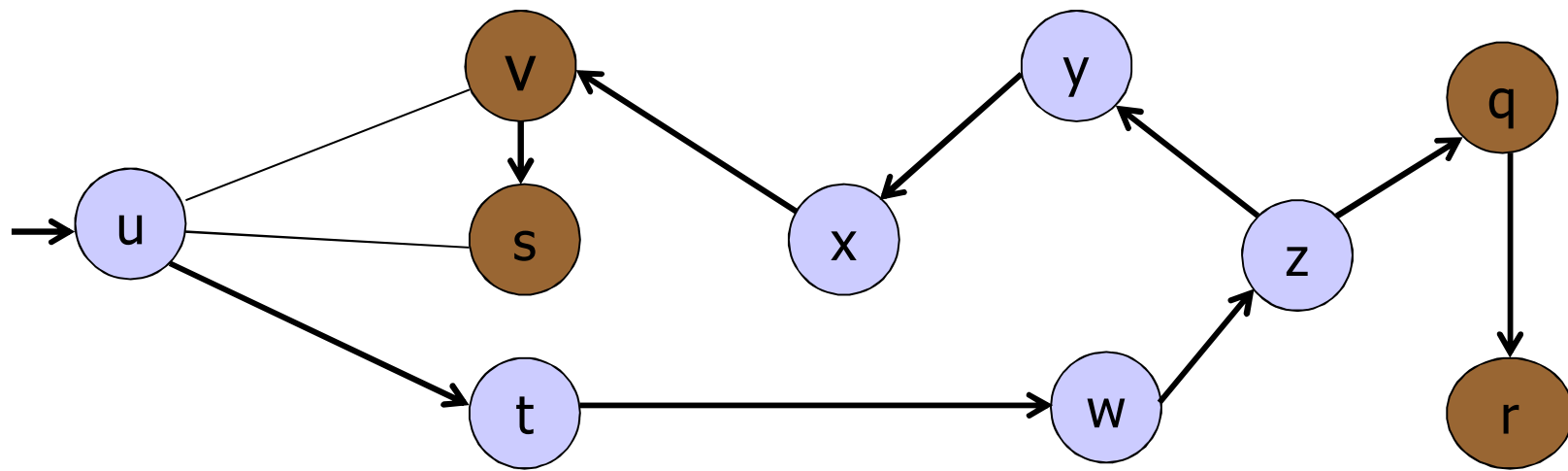
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh s



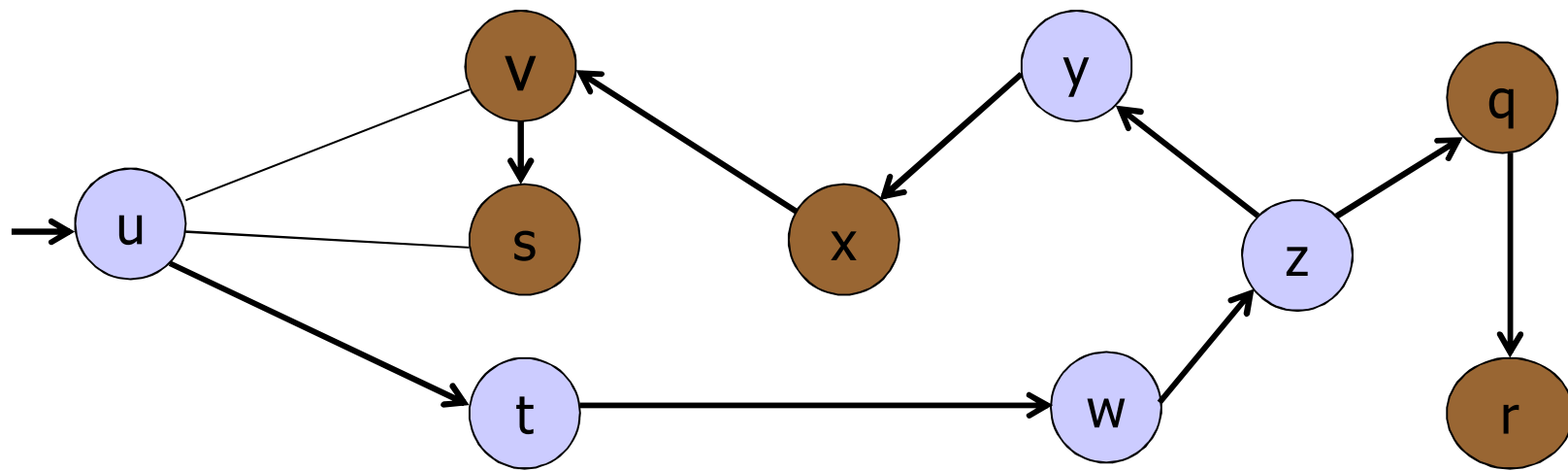
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh v



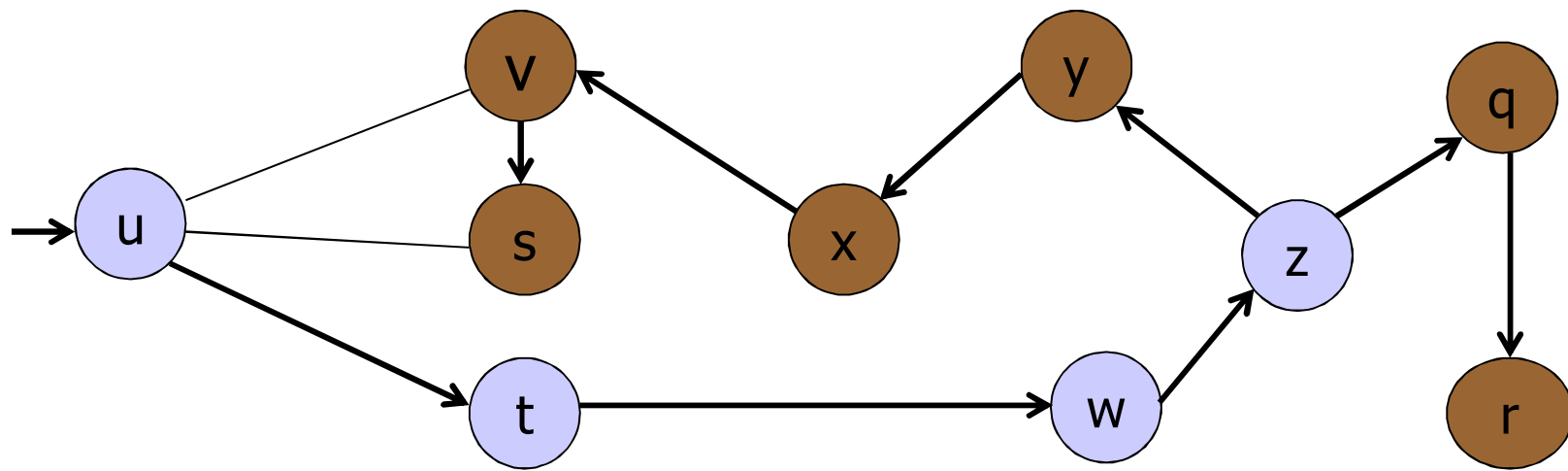
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh x



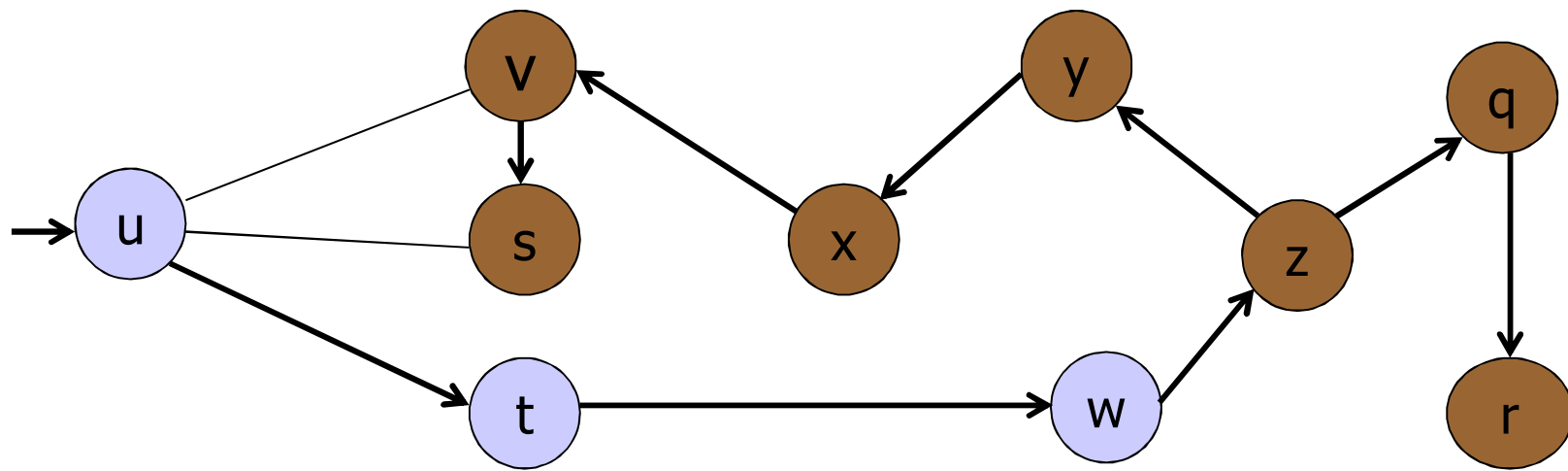
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh y



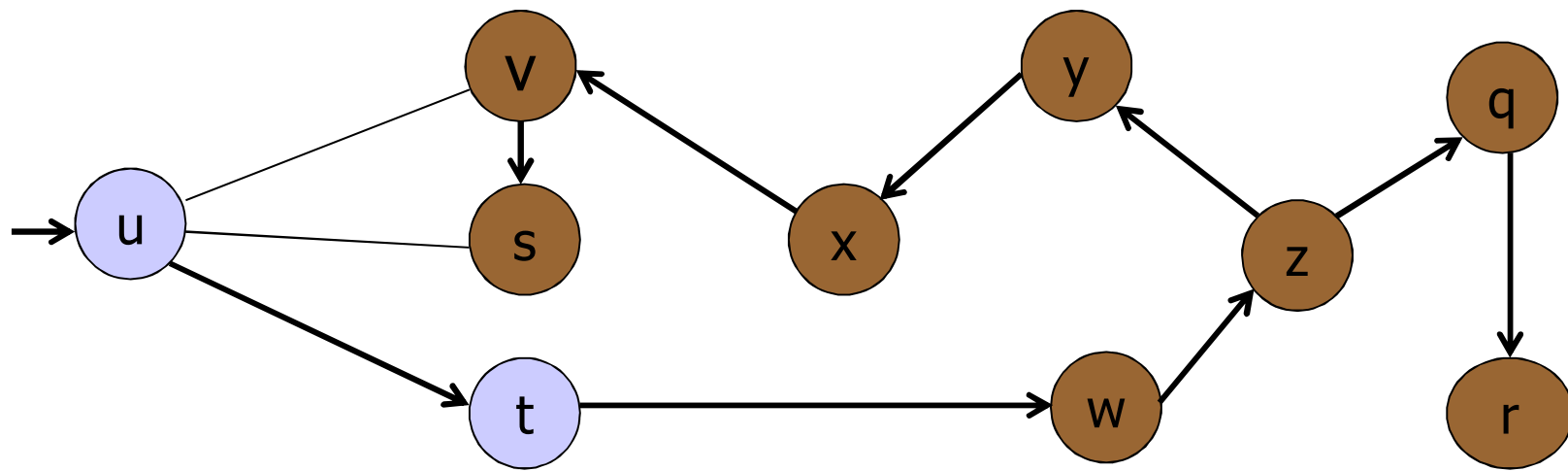
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh z



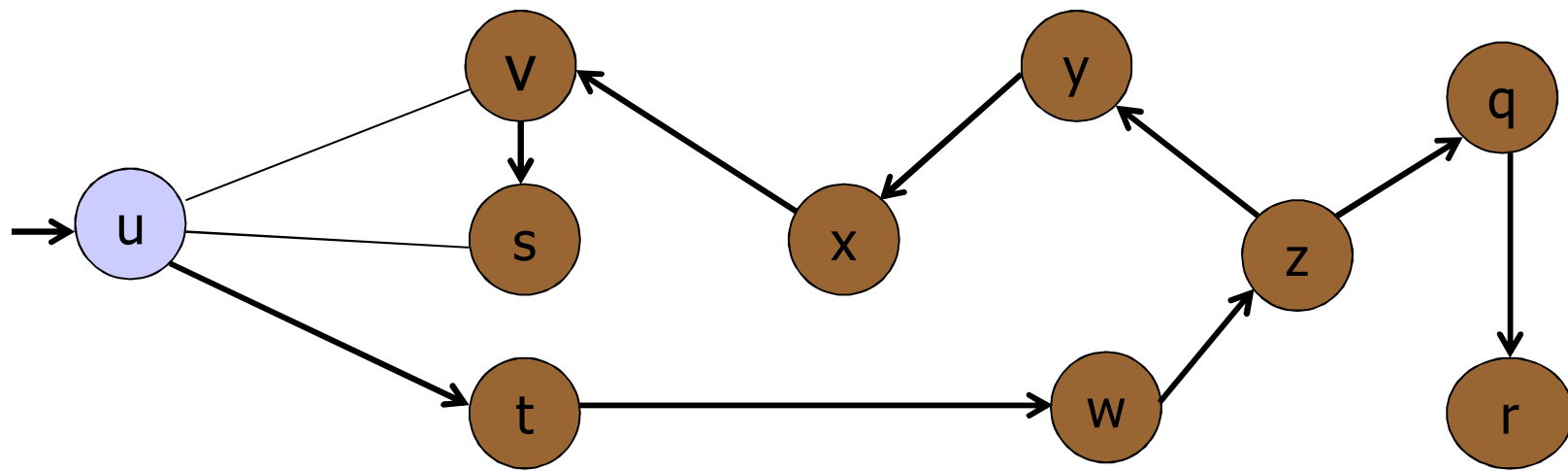
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh w



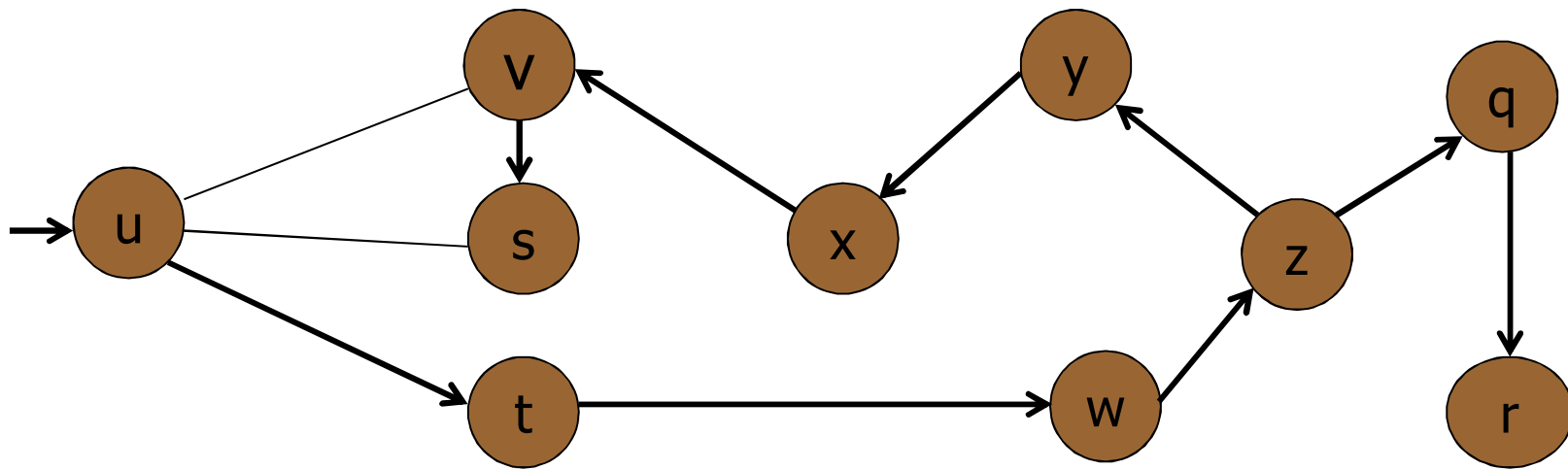
# THUẬT TOÁN DFS

Kết thúc tìm đỉnh t



# THUẬT TOÁN DFS

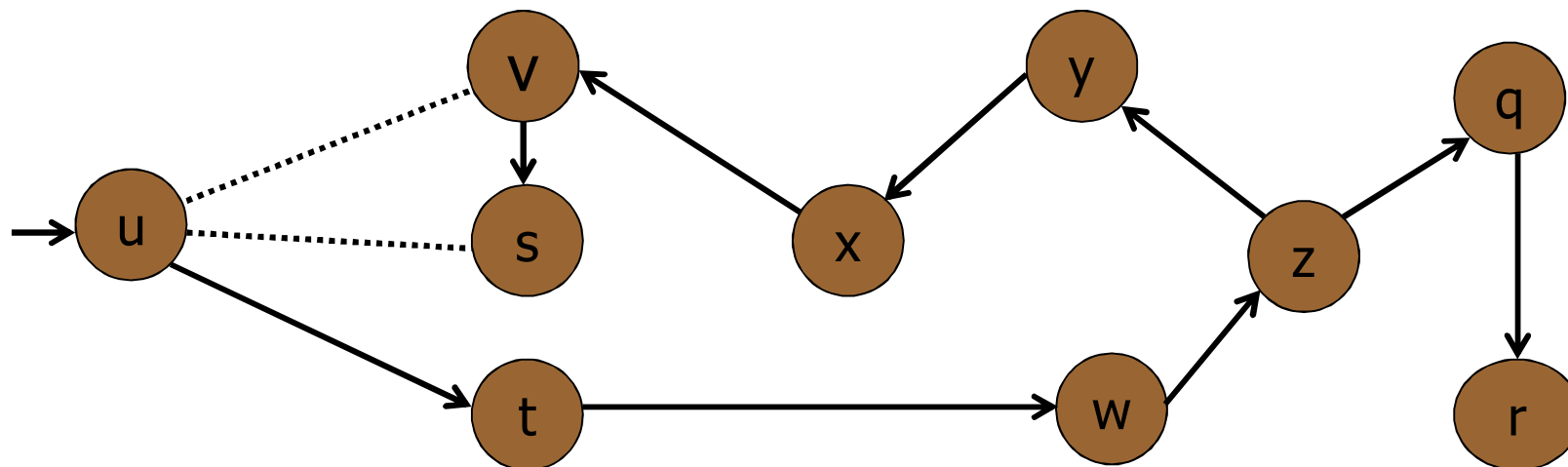
Kết thúc tìm đỉnh u





# THUẬT TOÁN DFS

Cây tìm kiếm theo chiều sâu (gốc u)



# THUẬT TOÁN DFS

DFS(G)

```
1 for each  $u \in V[G]$ 
2   do  $\text{color}[u] \leftarrow \text{white}$ 
3      $\pi[u] \leftarrow \text{NIL}$ 
4 for each  $u \in V[G]$ 
5   do if  $\text{color}[u] = \text{white}$ 
6     then DFS-VISIT( $u$ )
```

# THUẬT TOÁN DFS

DFS-VISIT(u)

```
1 color[u] ← gray
2 for each v ∈ Adj[u]
3     do if color[v]=white
4         then  $\pi[v] \leftarrow u$ 
5             DFS-VISIT(v)
6 color[u] ← brown;
```

# PHÂN TÍCH DFS

- Nếu chưa tính thời gian thực thi DFS-VISIT, vòng lặp 1-3 và 4-6 có chi phí là  $O(V)$
- Trong một lần thực thi DFS-VISIT( $u$ ), vòng lặp 2-5 thực thi trong  $|Adj[u]|$  lần
- Vì  $\sum_{u \in V} |Adj[u]| = O(E)$ , nên tổng chi phí thực thi dòng 4-6 của DFS-VISIT (trong DFS) là  $O(E)$ .
- Vậy thời gian chạy của DFS là  $O(V+E)$

# TÌM KIẾM THEO CHIỀU RỘNG (Breadth-First Search-BFS)

- Thuật toán BFS
- Phân tích BFS

# THUẬT TOÁN BFS

Ý tưởng thuật toán

- Bắt đầu tìm kiếm từ **đỉnh s** cho trước tùy ý
- Tại thời điểm đã tìm thấy  $u$ , thuật toán tiếp tục tìm kiếm tập tất cả các **đỉnh kề với  $u$**
- Thực hiện quá trình này cho các đỉnh còn lại

# THUẬT TOÁN BFS

Ý tưởng thuật toán

- Dùng một hàng đợi để duy trì trật tự tìm kiếm theo chiều rộng
- Dùng các màu để không lặp lại các đỉnh tìm kiếm
- Dùng một mảng để lưu trữ đỉnh đi trước của đỉnh được tìm kiếm

# THUẬT TOÁN BFS

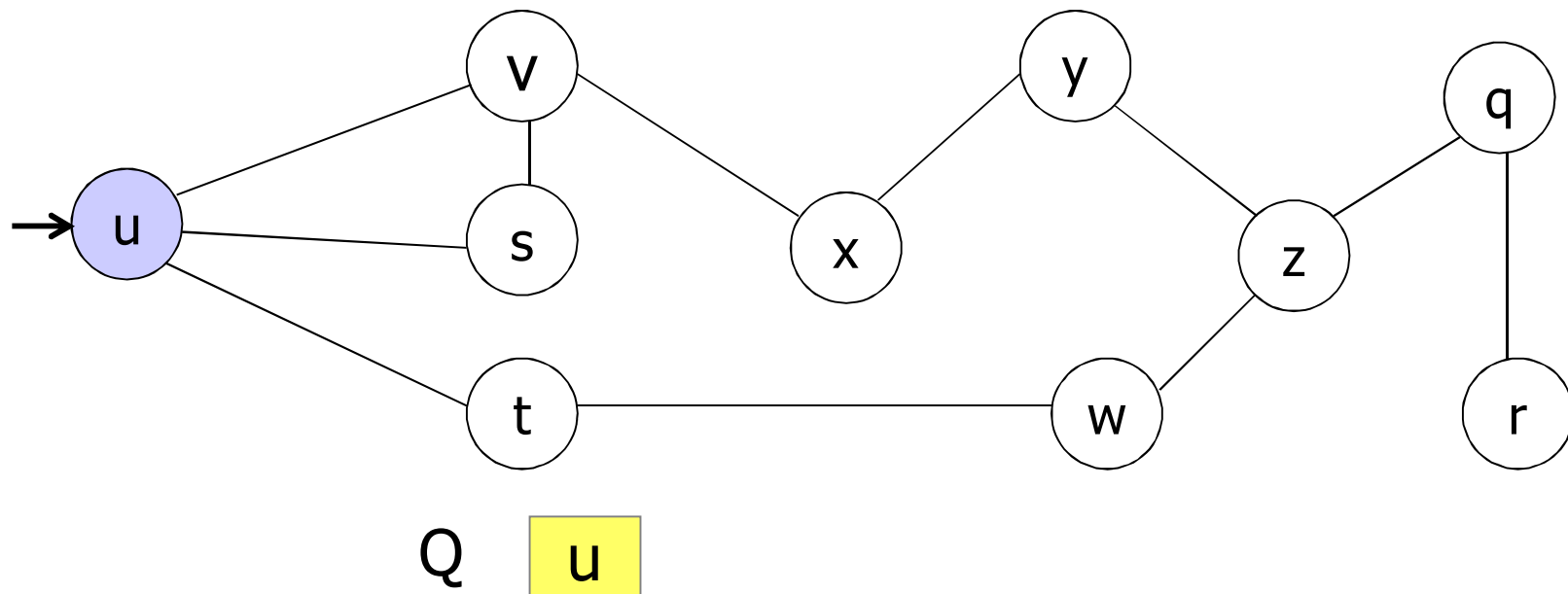
BFS( $G, s$ )

```
1  for each  $u \in V[G] - \{s\}$ 
2      do  $\text{color}[u] \leftarrow \text{white}$ 
3           $\pi[u] \leftarrow \text{NIL}$ 
4   $\text{color}[s] \leftarrow \text{gray};$ 
5   $\pi[s] \leftarrow \text{NIL};$ 
6   $Q \leftarrow \emptyset$ 
7  ENQUEUE( $Q, s$ )
8  while  $Q \neq \emptyset$ 
9      do  $u \leftarrow \text{DEQUEUE}(Q)$ 
10         for each  $v \in \text{Adj}[u]$ 
11             do if  $\text{color}[v] \leftarrow \text{white}$ 
12                 then  $\text{color}[v] \leftarrow \text{gray}$ 
13                      $\pi[v] \leftarrow u$ 
14                     ENQUEUE( $Q, v$ )
15          $\text{color}[u] \leftarrow \text{brown}$ 
```



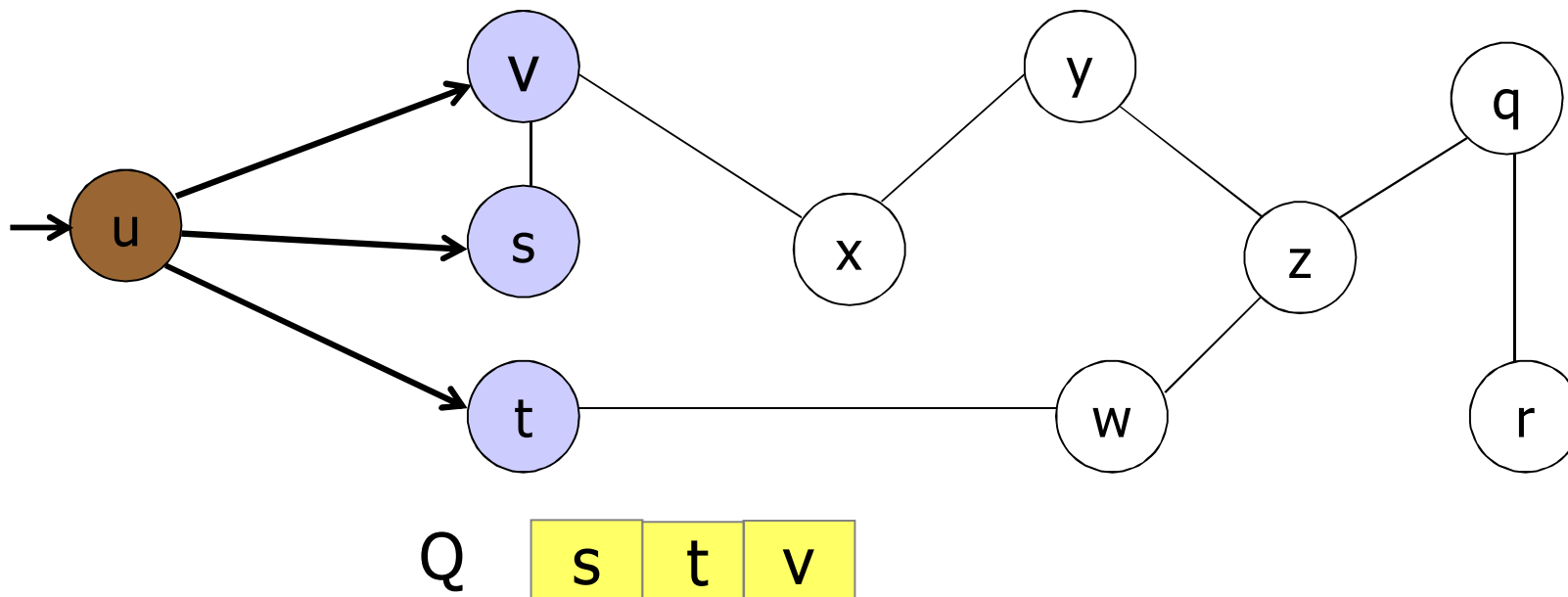
# THUẬT TOÁN BFS

Khởi đầu các đỉnh  $G$  có màu trắng (đỉnh xuất phát  $u$  màu xám)



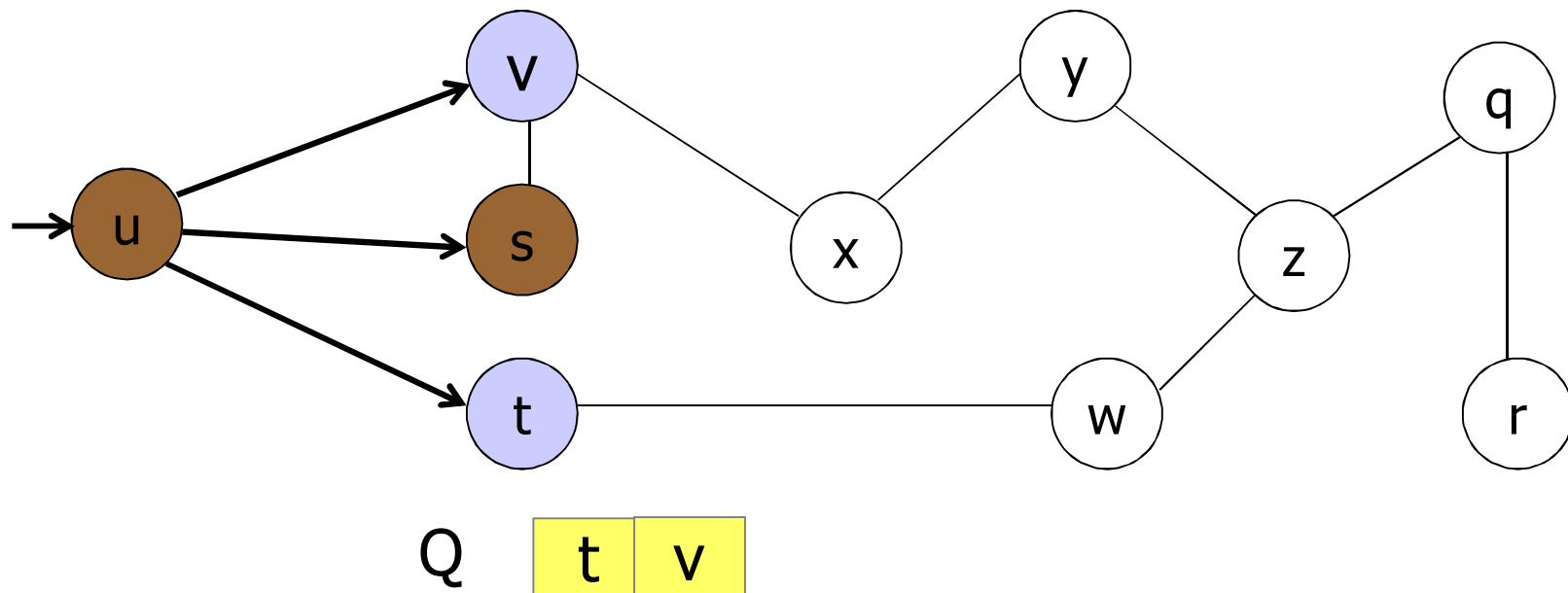
# THUẬT TOÁN BFS

Tìm thấy s, t, v và kết thúc tìm u (tô nâu)



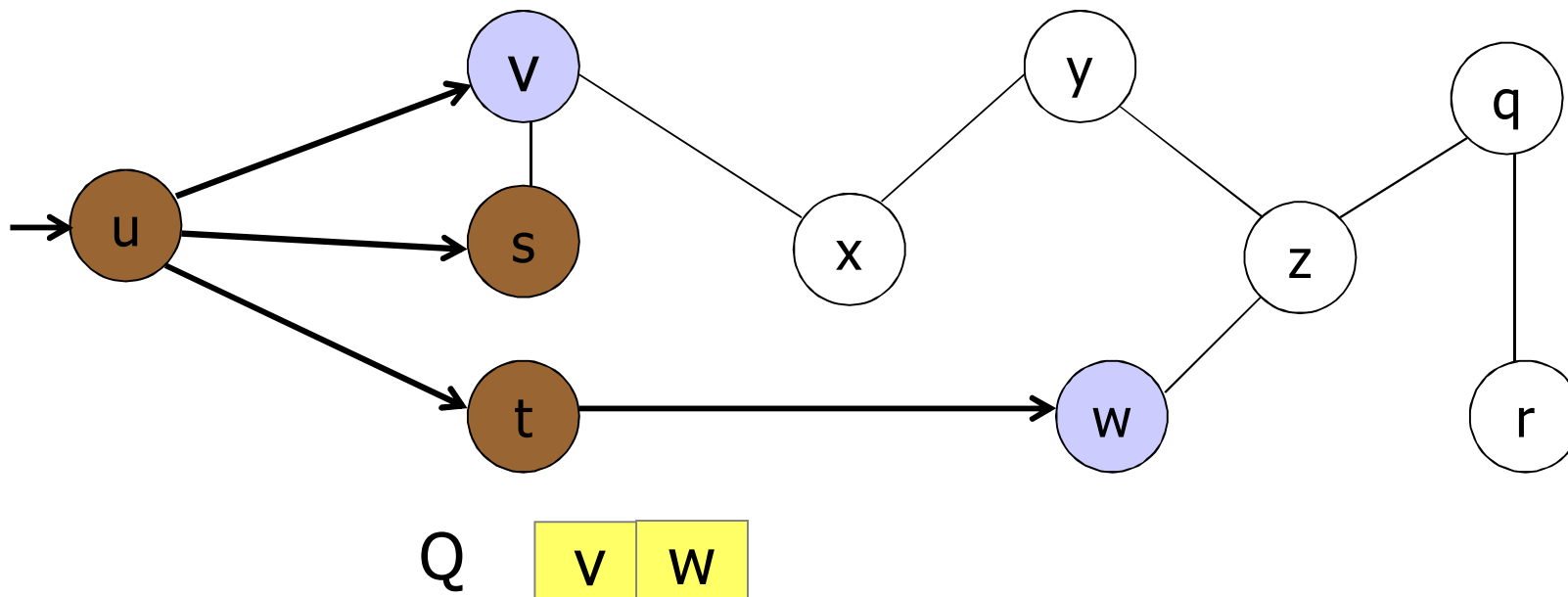
# THUẬT TOÁN BFS

Kết thúc tìm s (tô nâu)



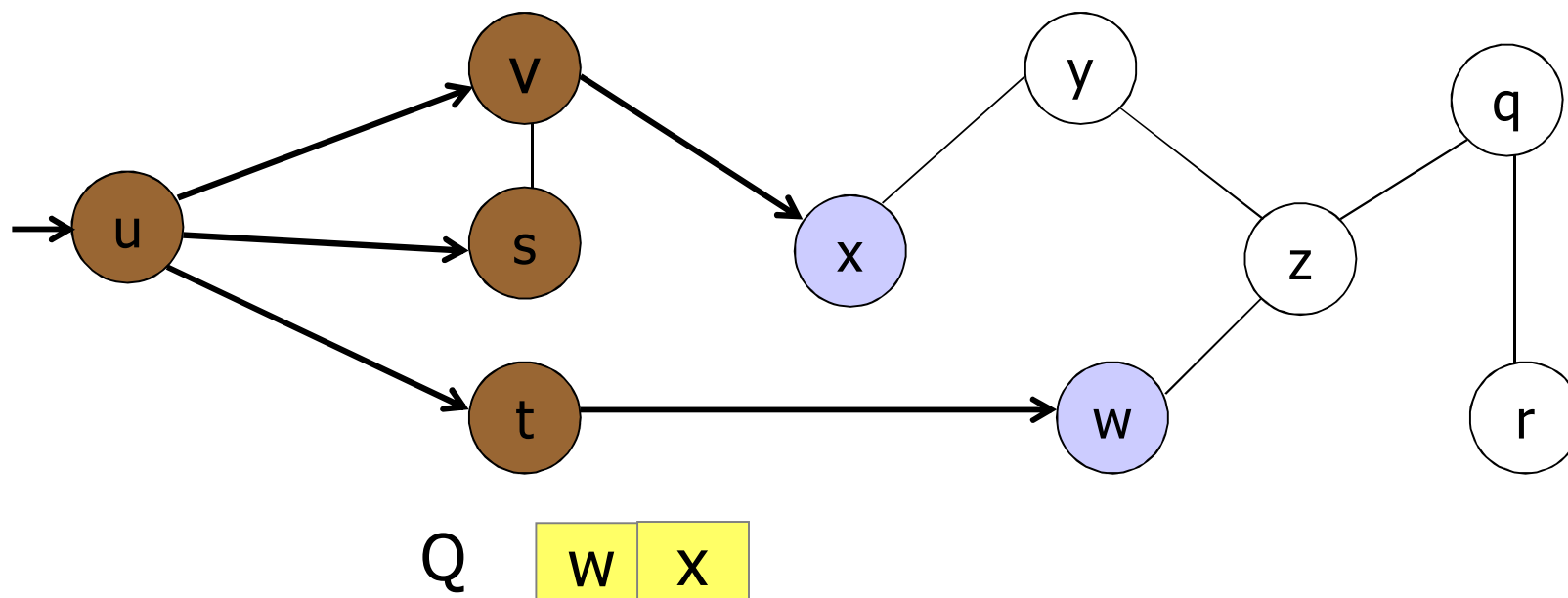
# THUẬT TOÁN BFS

Tìm thấy w, kết thúc tìm t (tô nâu)



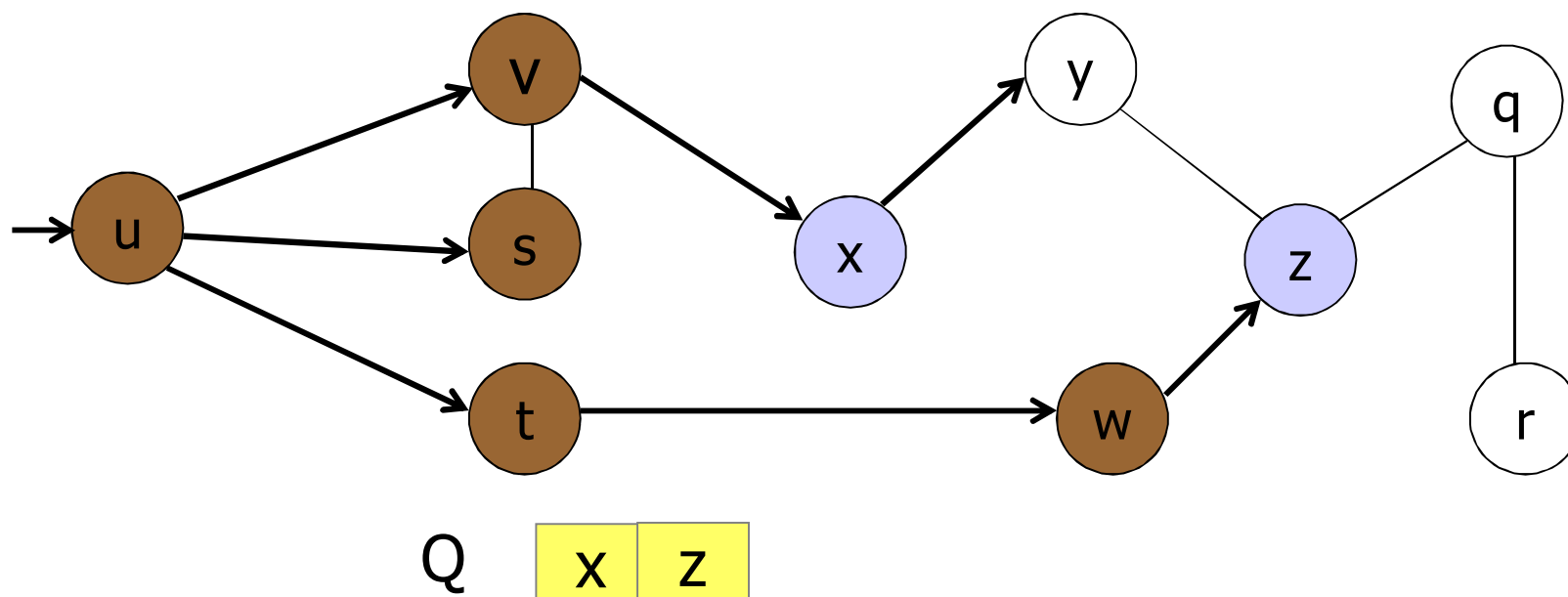
# THUẬT TOÁN BFS

Tìm thấy x, kết thúc tìm v (tô nâu)



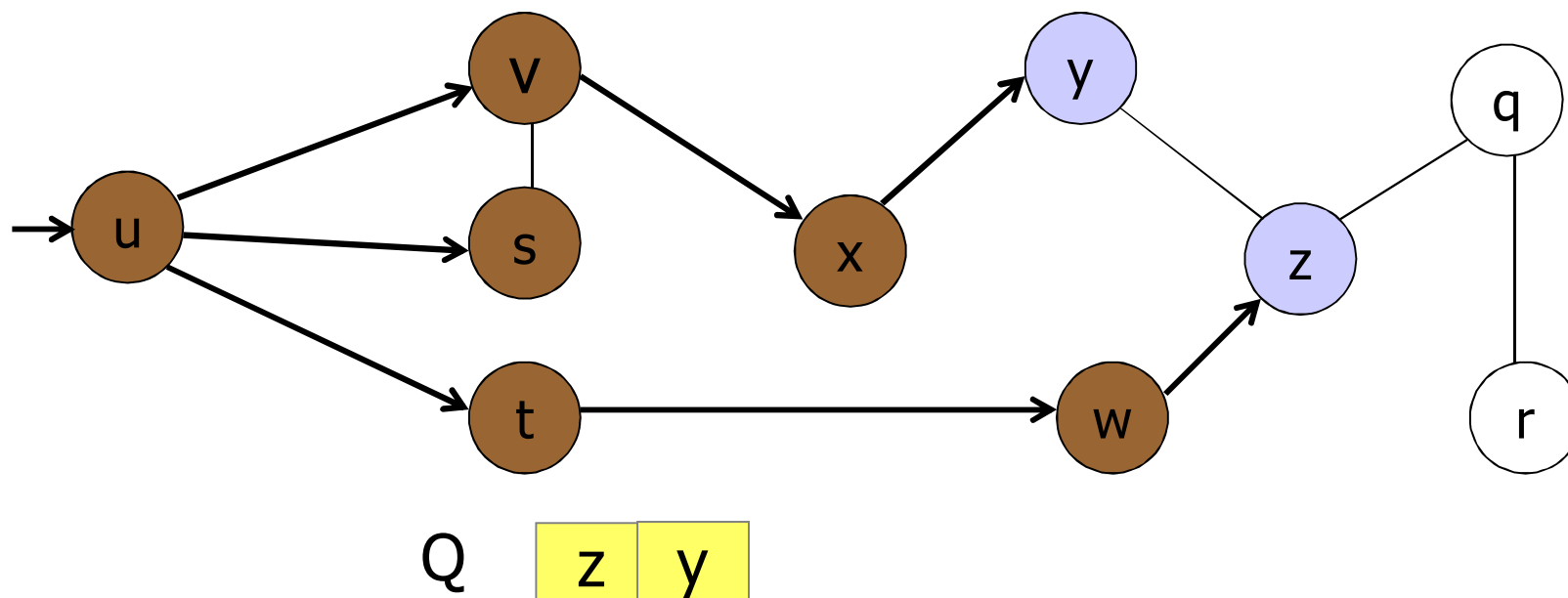
# THUẬT TOÁN BFS

Tìm thấy z, kết thúc tìm w (tô nâu)



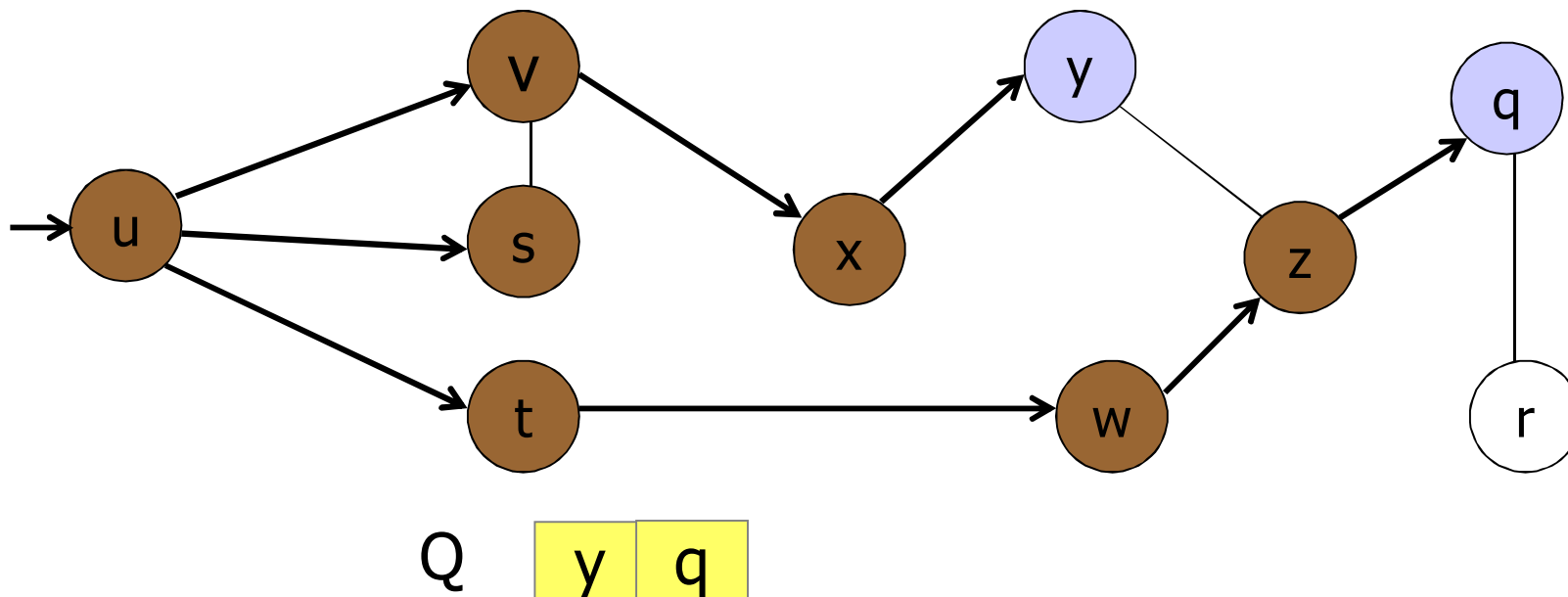
# THUẬT TOÁN BFS

Tìm thấy y, kết thúc tìm x (tô nâu)



# THUẬT TOÁN BFS

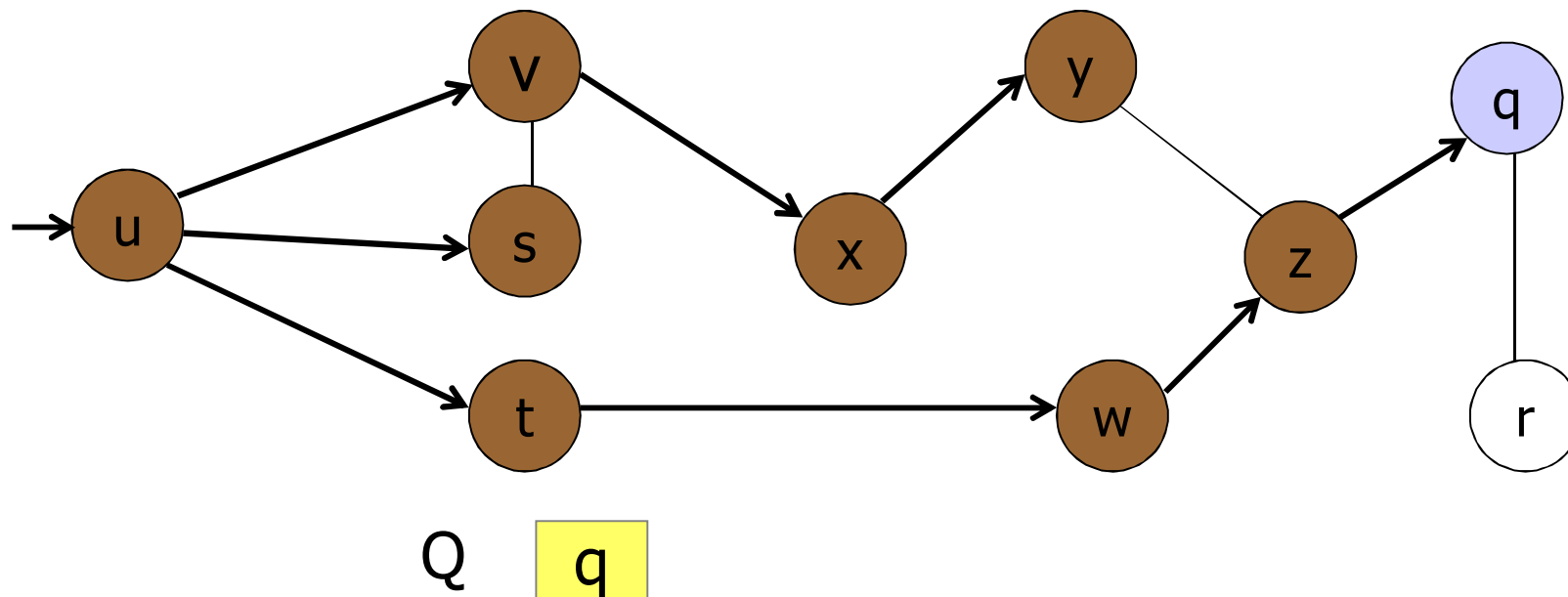
Tìm thấy q, kết thúc tìm z (tô nâu)





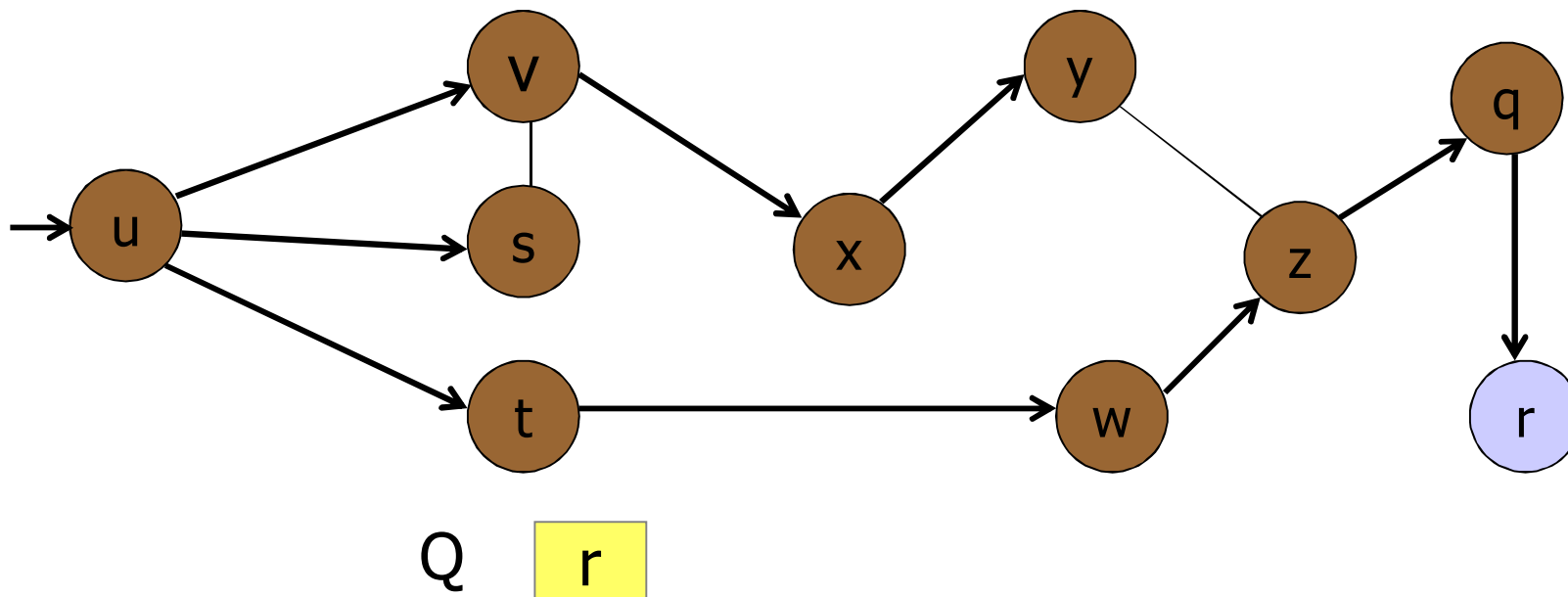
# THUẬT TOÁN BFS

Kết thúc tìm y (tô nâu)



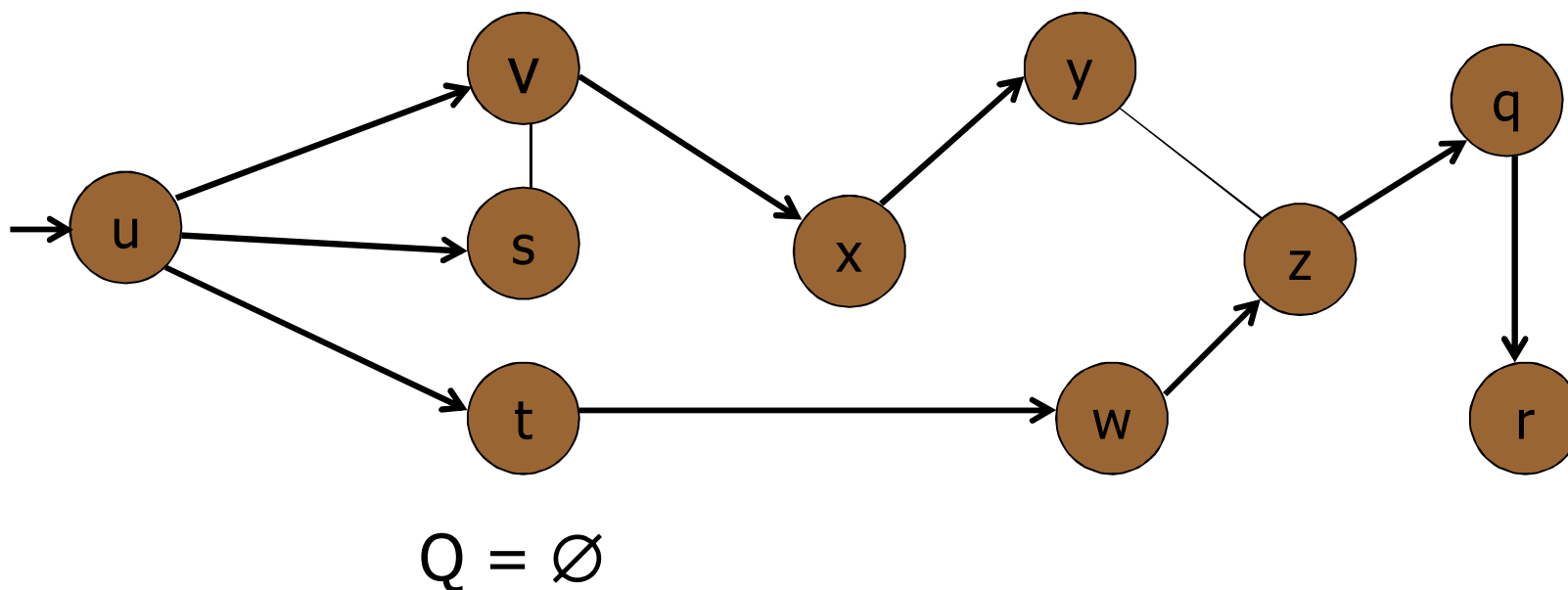
# THUẬT TOÁN BFS

Tìm thấy r, kết thúc tìm q (tô nâu)



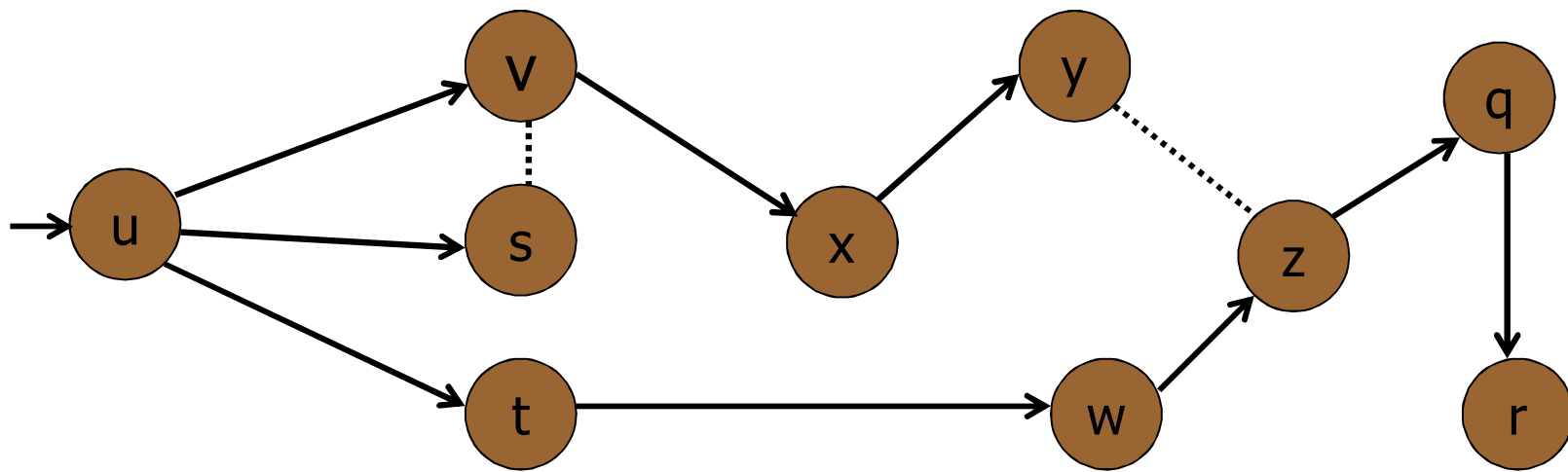
# THUẬT TOÁN BFS

Kết thúc tìm r (tô nâu), kết thúc tìm kiếm



# THUẬT TOÁN BFS

Cây tìm kiếm theo chiều rộng, gốc u



# PHÂN TÍCH BFS

- Tổng phí khởi tạo là  $O(V)$
- Mỗi thao tác trên hàng đợi là  $O(1)$ , vì vậy tổng thời gian cho thao tác trên hàng đợi là  $O(V)$
- Tổng thời gian chi phí cho quét các danh sách kề là  $O(E)$
- Tổng thời gian chạy của BFS là  $O(V+E)$

# MỘT SỐ ỨNG DỤNG

- Tìm đường đi giữa hai đỉnh
- Tính số thành phần liên thông
- Kiểm tra tính liên thông

# ĐƯỜNG ĐI GIỮA HAI ĐỈNH

PRINT-PATH( $G, s, v$ )

```
1  if  $v = s$ 
2      then print  $s$ 
3      else if  $\pi[v] = \text{NIL}$ 
4          then print “no path from”  $s$  “to”  $v$  “exists”
5          else PRINT-PATH( $G, s, \pi[v]$ )
6          print  $v$ 
```

# TÍNH THÀNH PHẦN LIÊN THÔNG

Connect\_Component(G)

```
1 for each  $u \in V[G]$ 
2     do color[u]  $\leftarrow$  white
3  $d \leftarrow 0$ 
4 for each  $u \in V[G]$ 
5     do if color[u] = white
6         then DFS-Visit(u)
7              $d \leftarrow d+1$ 
8 return d           //d là số thành phần liên thông
```



# KIỂM TRA TÍNH LIÊN THÔNG

Graph\_Connect(G)

```
1 for each  $u \in V[G]$ 
2   do color[u]  $\leftarrow$  white
3  $d \leftarrow 0$ 
4 for each  $u \in V[G]$ 
5   do if color[u] = white
6     then DFS-VISIT(u)
7            $d \leftarrow d+1$ 
8 if  $d = 1$ 
9   then return true
10  else return false
```