

Tema 8: Grafos

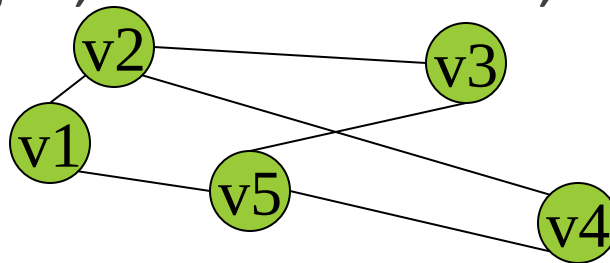
ESTRUCTURAS DE DATOS

Contenidos

- Introducción
- Definiciones
- Representación
- Recorridos

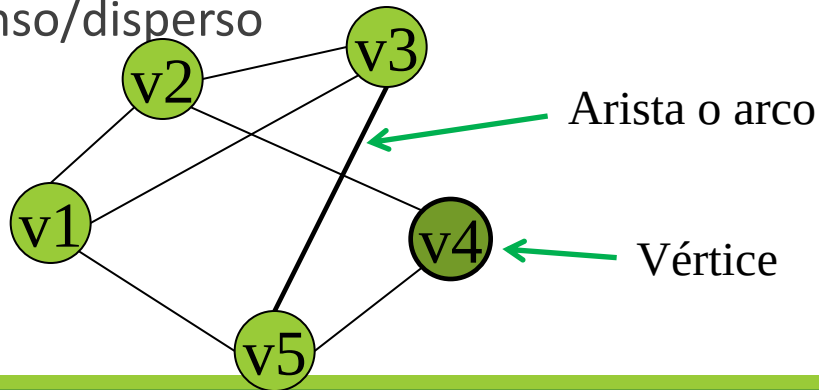
Introducción

- Los árboles se utilizaban para modelar jerarquías
- Los grafos son extensiones que se utilizan para modelar sistemas posiblemente menos jerárquicos o relaciones arbitrarias
- Por ejemplo caminos entre ciudades, redes de ordenadores, conectividades de píxeles en imágenes, optimización de viajes, tráfico urbano,...



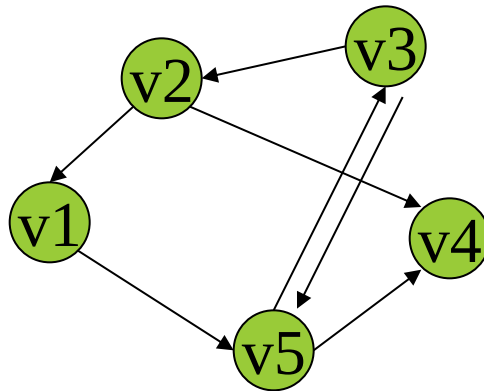
Definiciones

- Grafo: $G = (V, A)$
- V = Conjunto de vértices pertenecientes al grafo $G \{v1, v2, v3, \dots\}$
- A = Conjunto de pares $(v1, v2)$ donde $v1$ y $v2$ pertenecen a V . Representan los arcos (aristas) de G
- Densidad: relación entre #arcos vs #vértices
 - Grafo denso/disperso



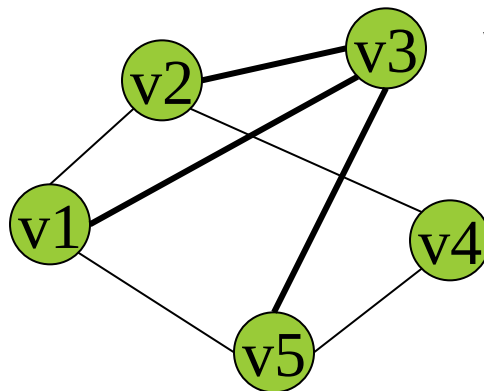
Definiciones

- Grafo dirigido o Digrafo: Cuando los arcos son ordenados $(v1, v2) \neq (v2, v1)$
 - Arcos “con sentido” $\rightarrow$ Hay “flechas”
 - Puede haber 2 arcos entre 2 nodos para indicar vecindad mutua (adyacentes)



Definiciones

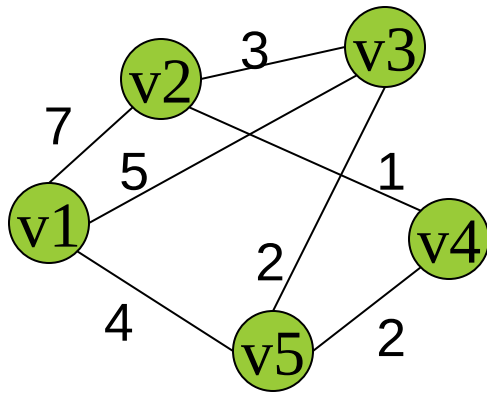
- **Adyacencia:** Dos vértices son adyacentes si existe un arco que los una (para grafos dirigidos: “adyacentes a”)



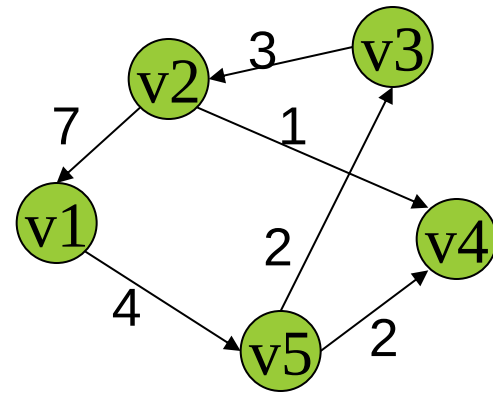
V3 es adyacente de V2, V1 y V5

Definiciones

- Grafo valorado o ponderado: Grafo cuyos arcos tienen asociados un peso o coste.



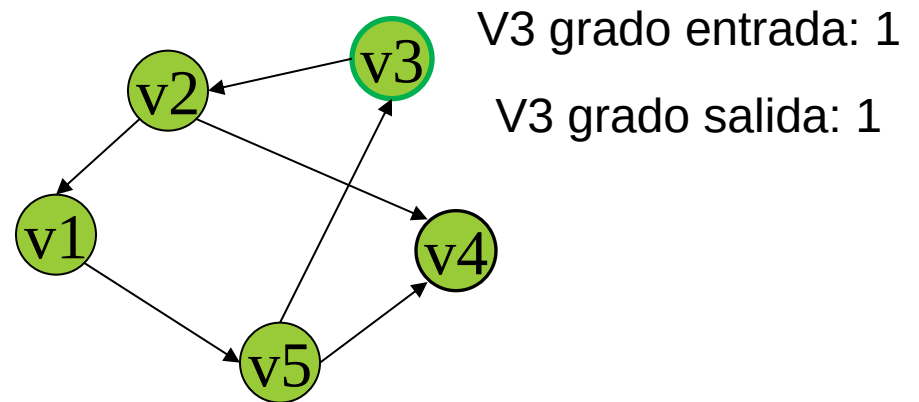
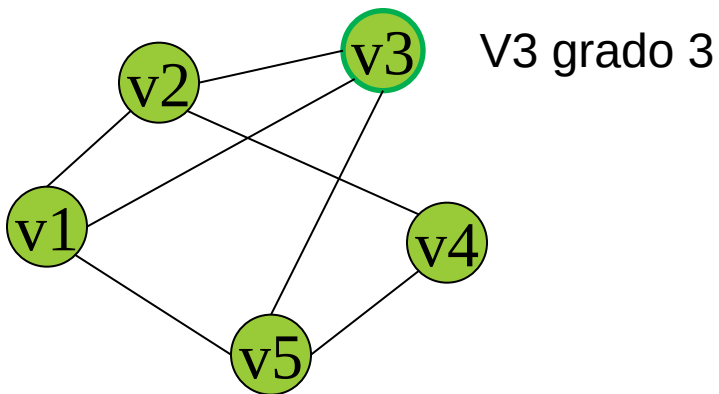
Grafo ponderado



Digrafo ponderado

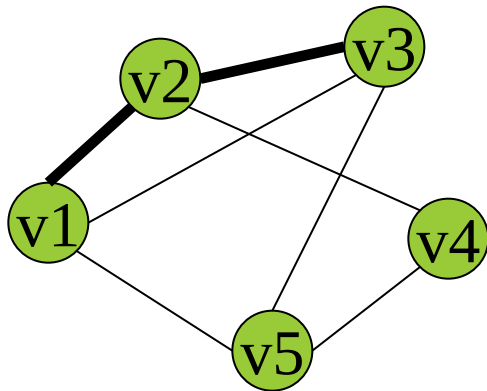
Definiciones

- Grado: Número de arcos que contienen a un nodo
- Grado de entrada y grado de salida:
 - Para grafos dirigidos (#arcos de salida/entrada)
- Nodos fuente/sumidero:
 - CON arcos SÓLO de salida/entrada



Definiciones

- Camino: Secuencia de vértices que hay que atravesar para llegar de un vértice a otro



Camino de V1 a V3:

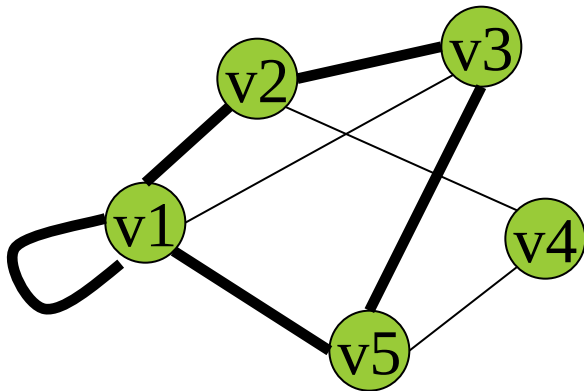
$(V1, V2), (V2, V3)$

Longitud: 2

- Longitud: Número de arcos que atraviesa un camino
- Camino simple: camino con vértices diferentes

Definiciones

Ciclo: posible camino de “v” a “v” (pasando por otros)



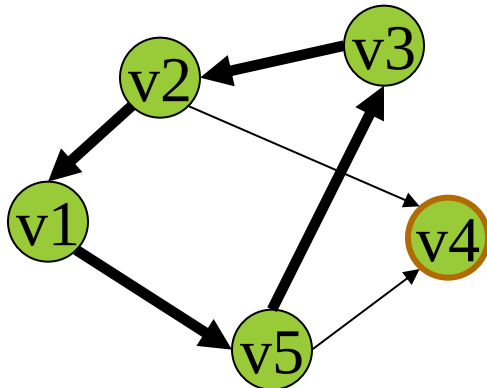
Ciclo a V1:

(**v1**, v2), (v2, v3), (v3, v5), (v5, **v1**)

- Bucle: camino de “v” a “v”

Definiciones

- Circuito: idem ciclo pero para grafos dirigidos



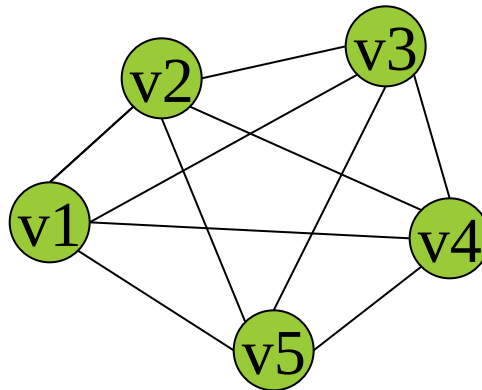
Circuito a V1:

$\langle \mathbf{v1}, v5 \rangle, \langle v5, v3 \rangle, \langle v3, v2 \rangle, \langle v2, \mathbf{v1} \rangle$

Definiciones

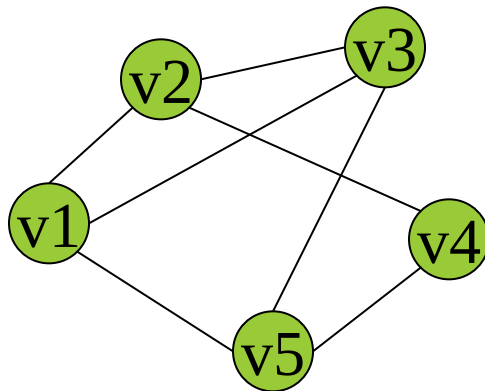


- Grafo completo: si existe una arista entre cada par de vértices
 - Máxima densidad

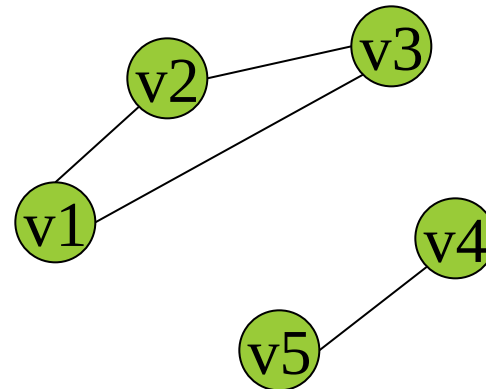


Definiciones

- Grafo conexo: Para grafo no dirigido si existe un camino desde un vértice a cualquier otro



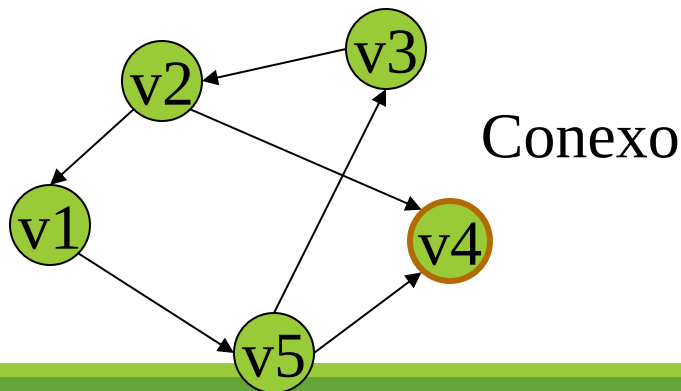
Conexo



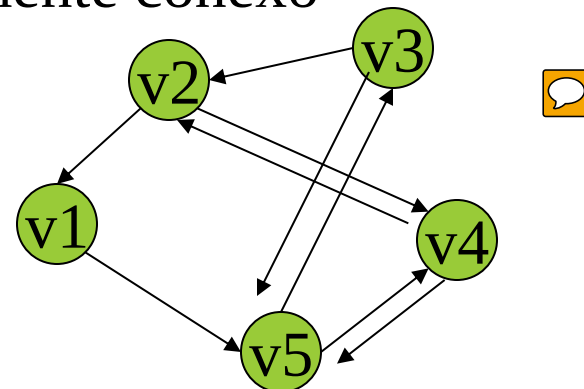
Inconexo
(2 componentes conexas)

Definiciones

- Grafo fuertemente conexo: Para un grafo dirigido, si existe un camino desde un vértice a cualquier otro
- Un grafo dirigido puede ser conexo (si lo es sin considerarlo dirigido) o fuertemente conexo



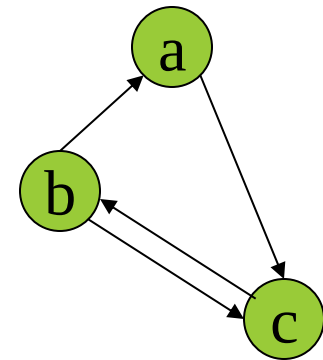
Fuertemente conexo



Representación

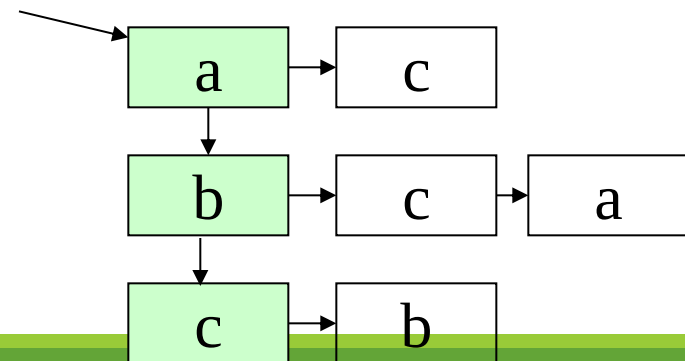
- Representación estática (grafo dirigido)
 - Conjunto nodos + Matriz de adyacencia

$$V = (a, b, c) \quad A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



- Representación dinámica (grafo dir.)
 - Lista de adyacencia
 - Incluye conjunto nodos

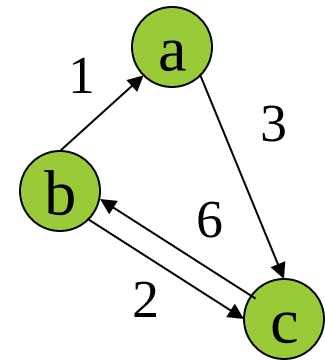
Especialmente útil si
el grafo es disperso



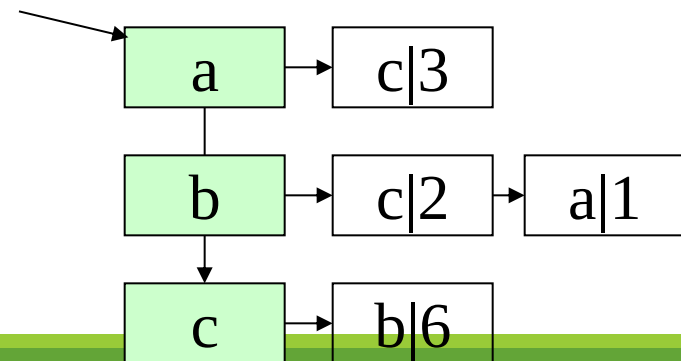
Representación

- Representación estática grafo ponderado
 - Conjunto nodos + Matriz de adyacencia con pesos

$$V = (a, b, c) \quad A = \begin{pmatrix} 0 & 0 & 3 \\ 1 & 0 & 2 \\ 0 & 6 & 0 \end{pmatrix}$$



- Repr. dinámica grafo ponderado
 - Lista de adyacencia 
 - Nodos con campo peso



Recorridos

- Recorrer un grafo (*graph traversal*): visitar todos los nodos alcanzables a partir de uno dado (¡sin repetir nodos!)
 - Recorrido en anchura (BFS)
 - Recorrido en profundidad (DFS)
 - Múltiples variantes para determinados problemas y aplicaciones
 - Algoritmos Prim, Dijkstra, Bellman-Ford, Kruskal,...

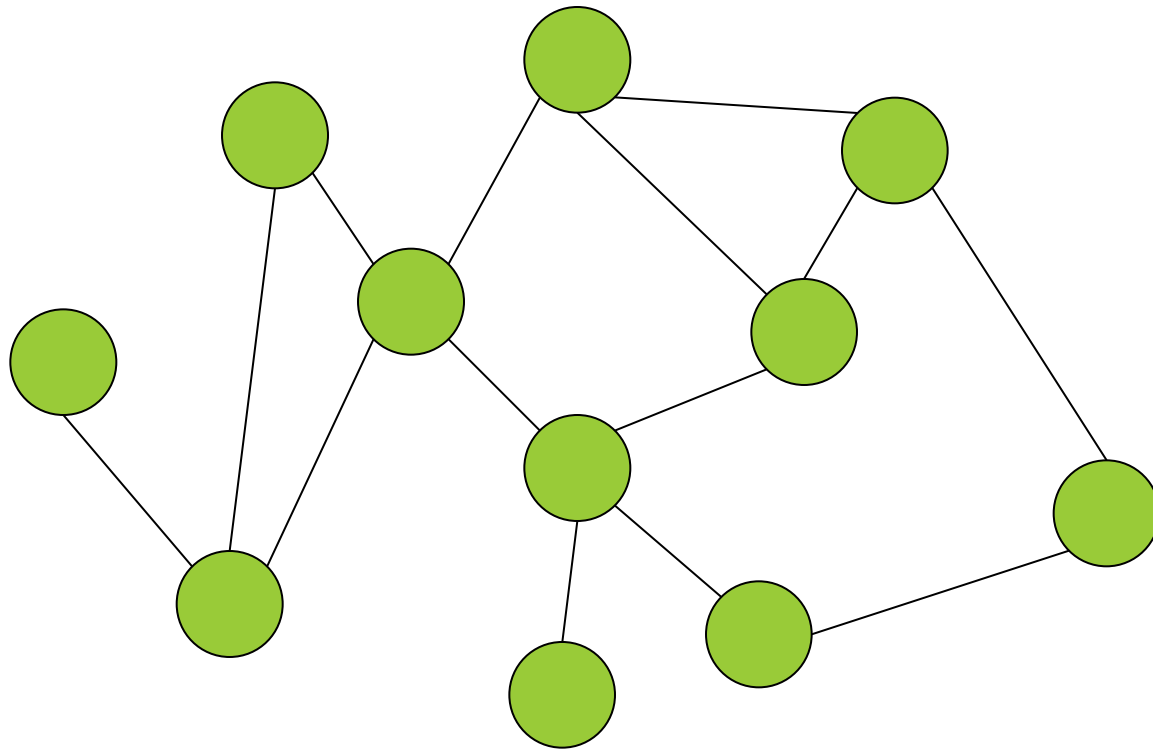
Recorridos

- De forma iterativa se necesitan ciertas estructuras de datos auxiliares
 - Conjunto “visitados” para no repetir nodos y entrar en ciclos (*loop* infinito)
 - Guarda elementos ya visitados
 - Estructura FIFO (*anchura/breadth-first/BFS*) o pila (*profundidad/depth-first/DFS*) por la que pasan los nodos para guardar memoria y orden de visita
 - Guardan direcciones de nodos (punteros) que quedan por visitar

Prototipo:

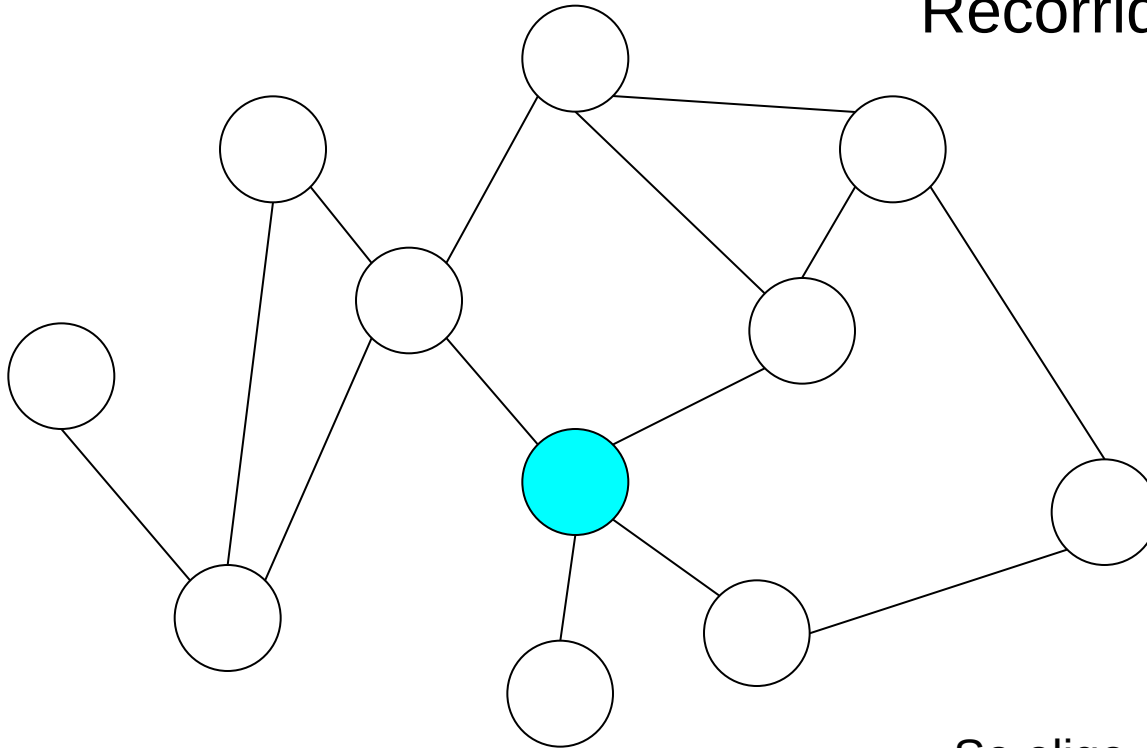
```
PROCEDURE Recorrido(g: TGrafo; origen: TElemento; VAR l: TLista)
```

Recorridos



Recorridos

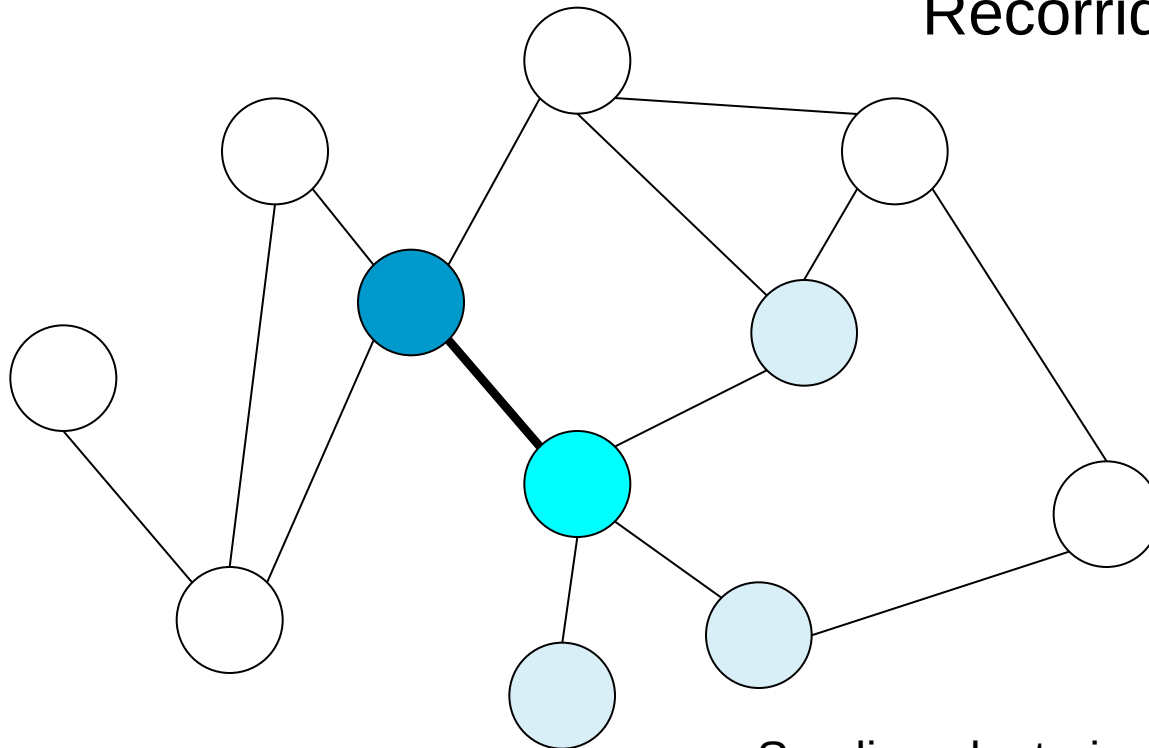
Recorrido en anchura



Se elige origen del recorrido

Recorridos

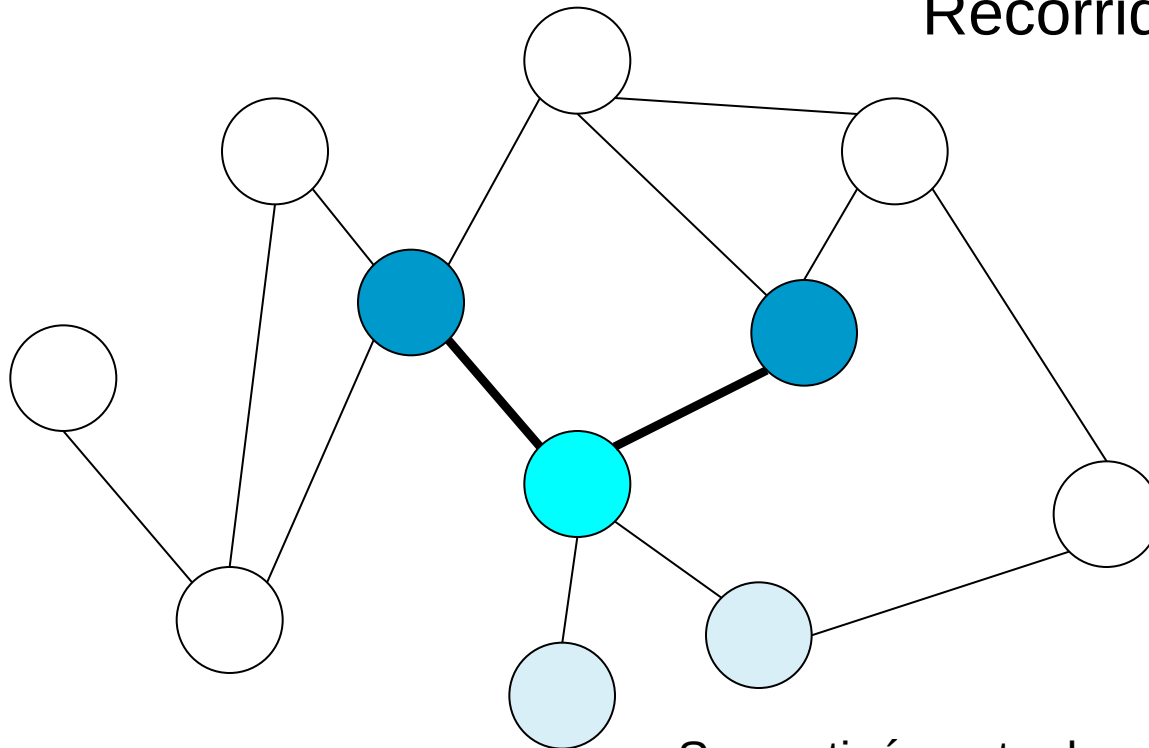
Recorrido en anchura



Se elige aleatoriamente siguiente a visitar
entre sus nodos vecinos

Recorridos

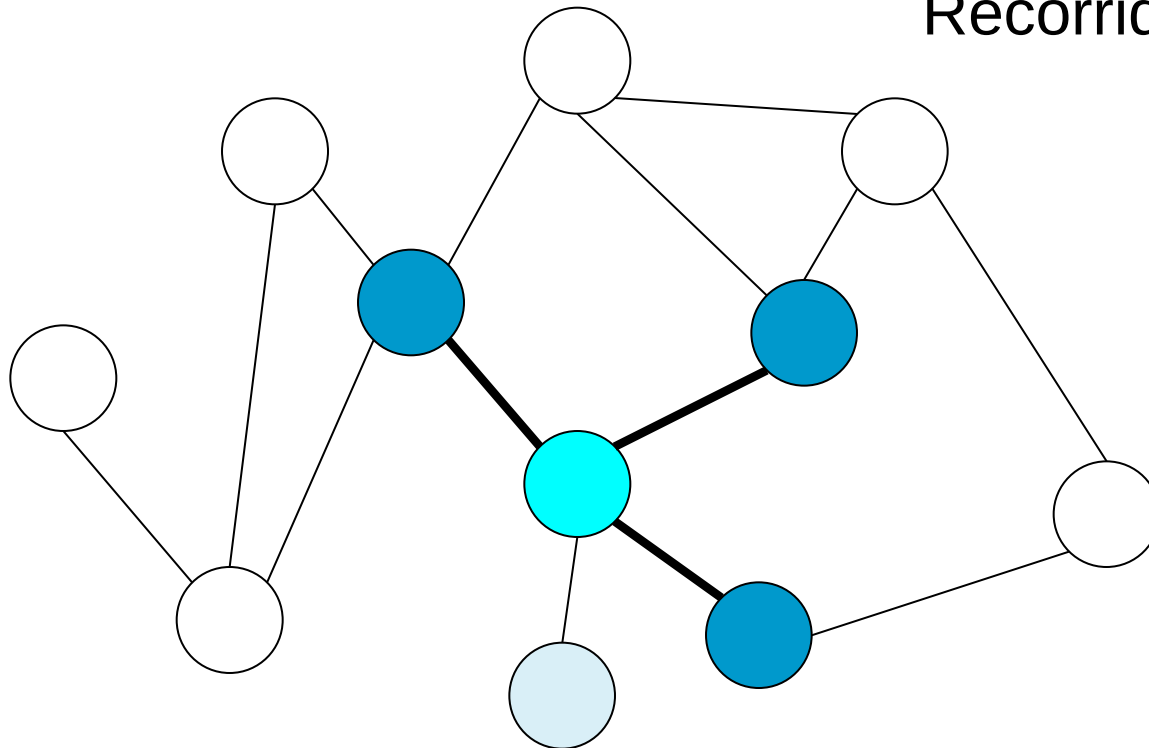
Recorrido en anchura



Se continúa entre los restantes vecinos del origen (si quedan sin visitar)

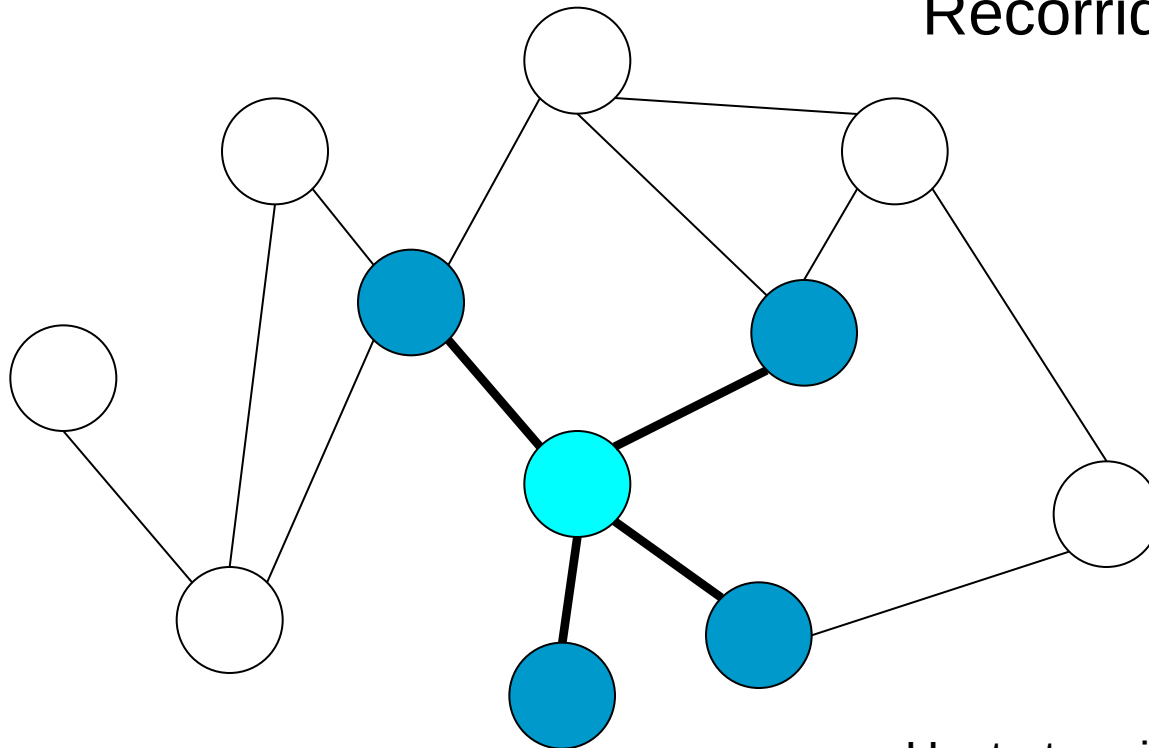
Recorridos

Recorrido en anchura



Recorridos

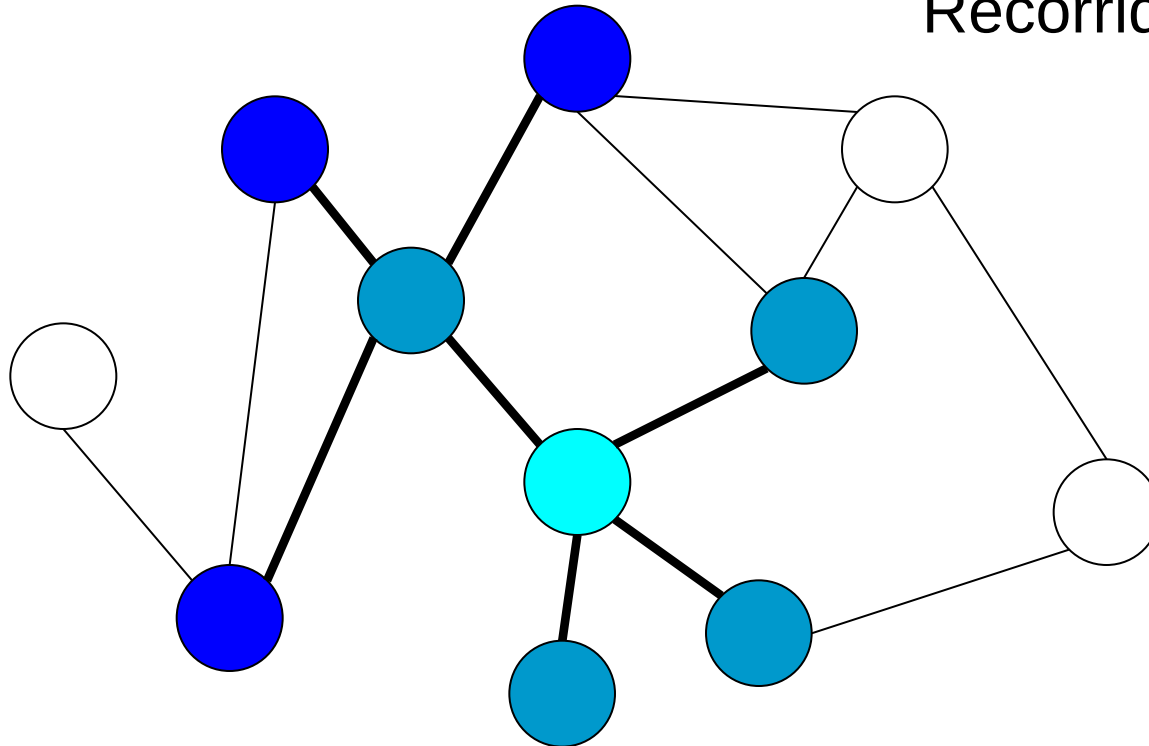
Recorrido en anchura



Hasta terminar con la vecindad 1

Recorridos

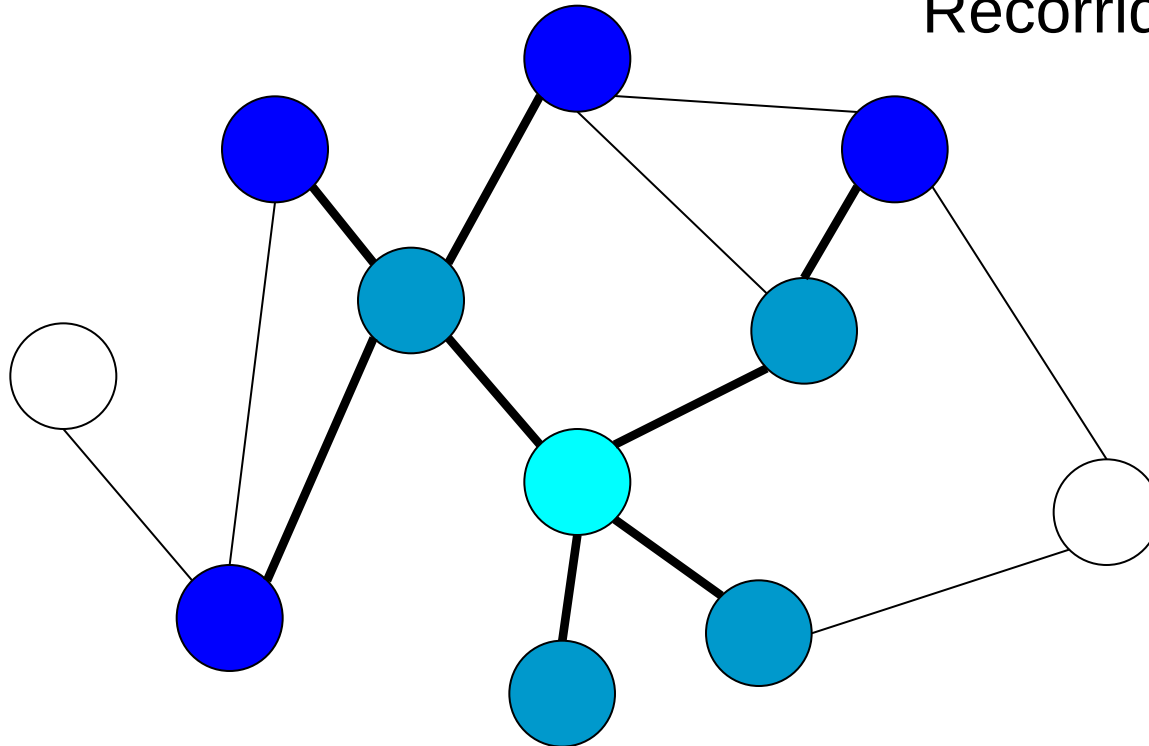
Recorrido en anchura



Se elige siguiente nodo desde el que
continuar el recorrido (vecindad 2)

Recorridos

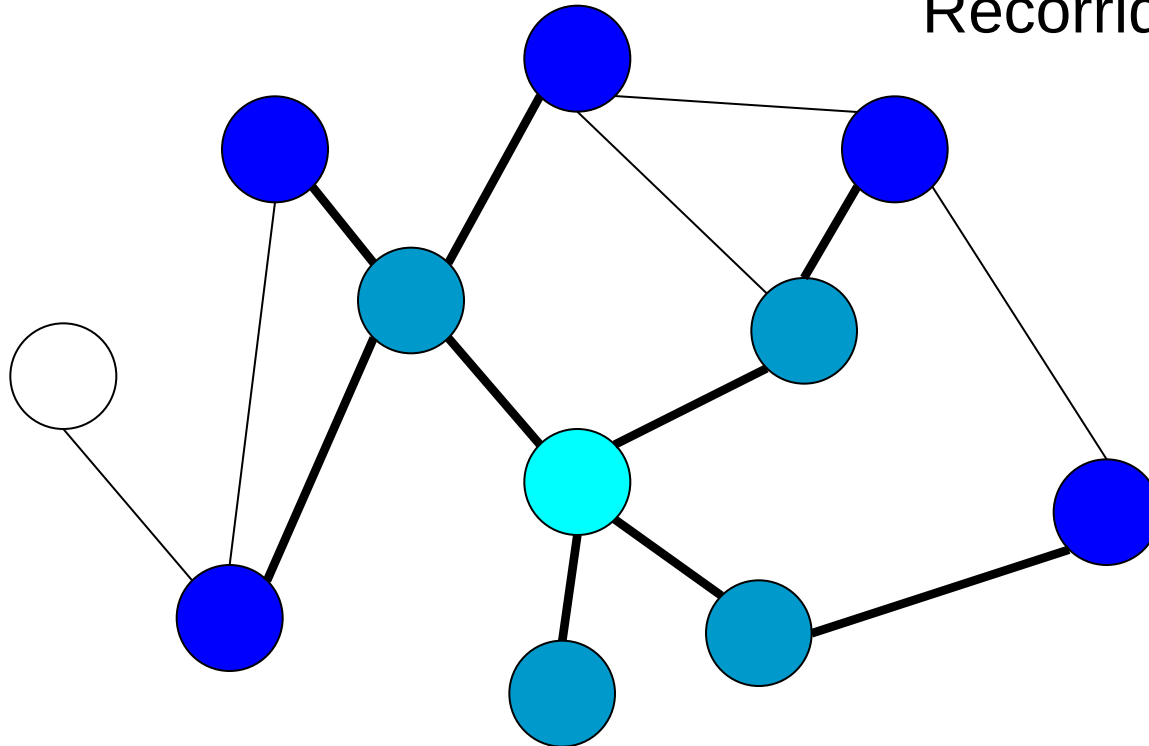
Recorrido en anchura



Se visitan otros vecinos de la vecindad 1

Recorridos

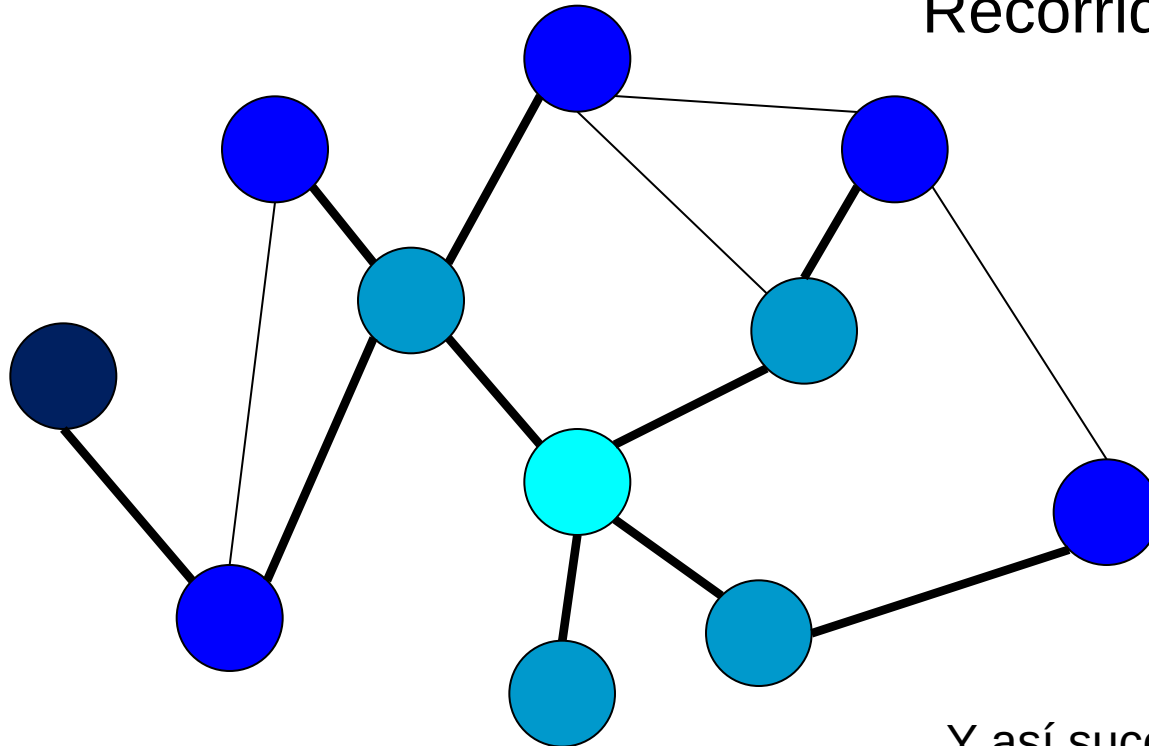
Recorrido en anchura



Hasta terminar con la vecindad 2

Recorridos

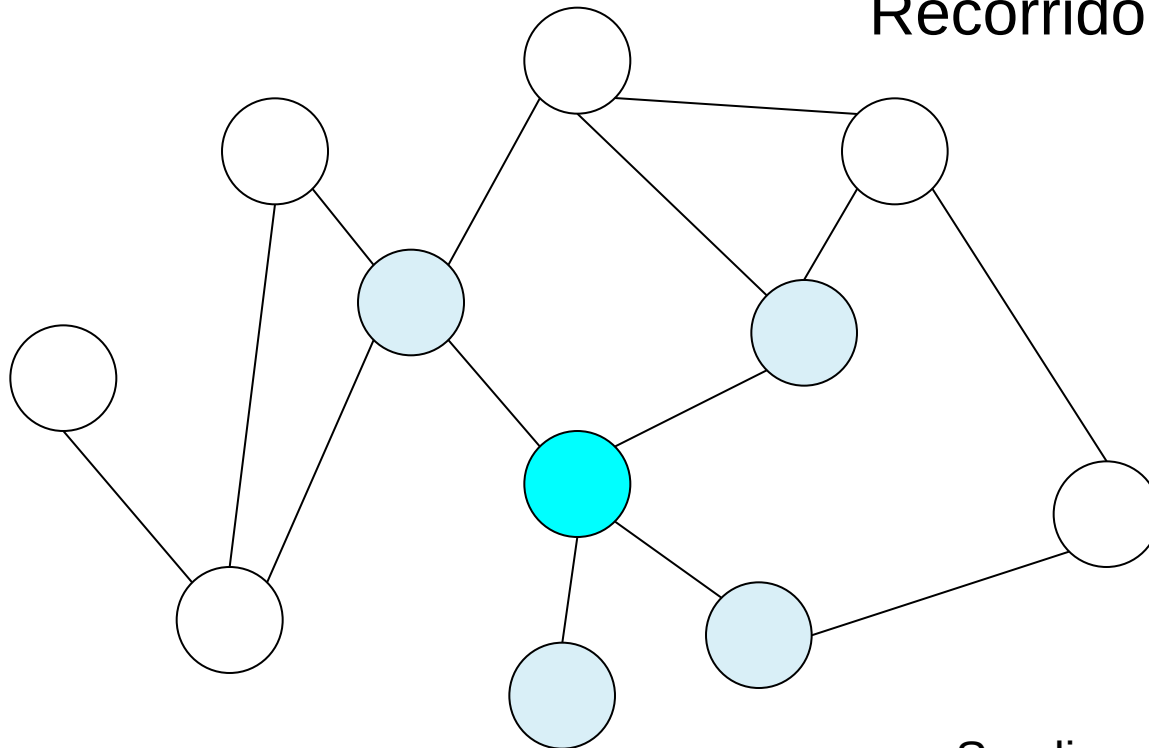
Recorrido en anchura



Y así sucesivamente hasta
terminar con todos los nodos

Recorridos

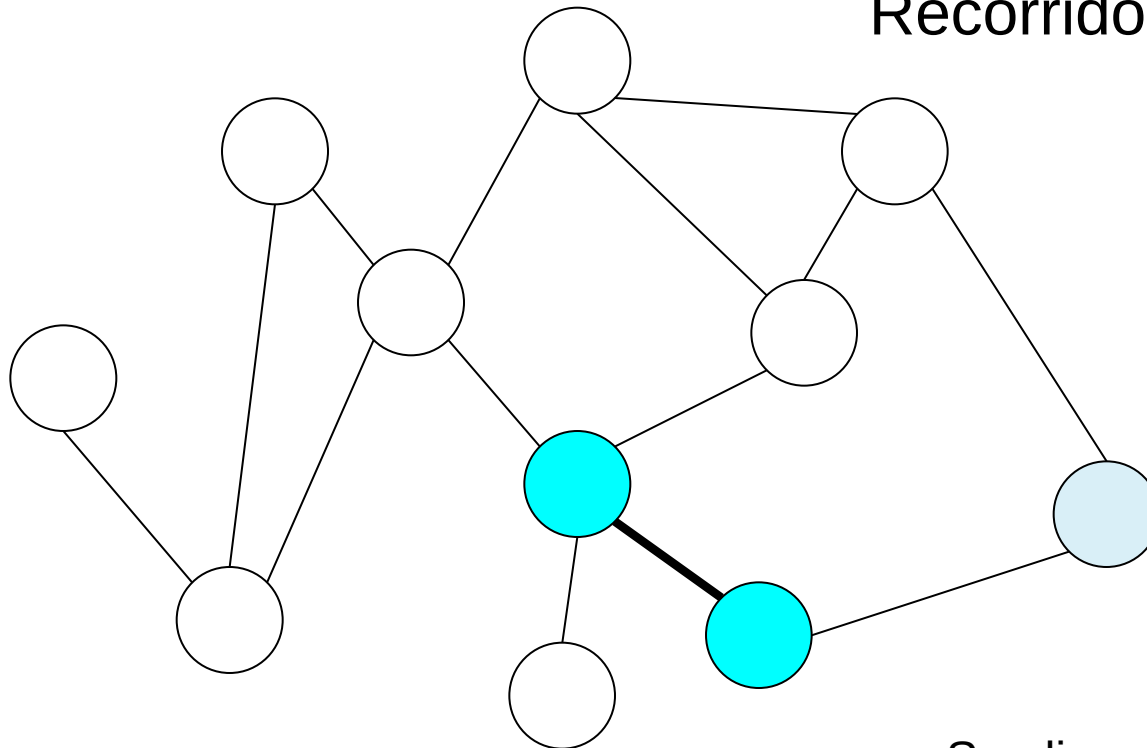
Recorrido en profundidad



Se elige origen del recorrido

Recorridos

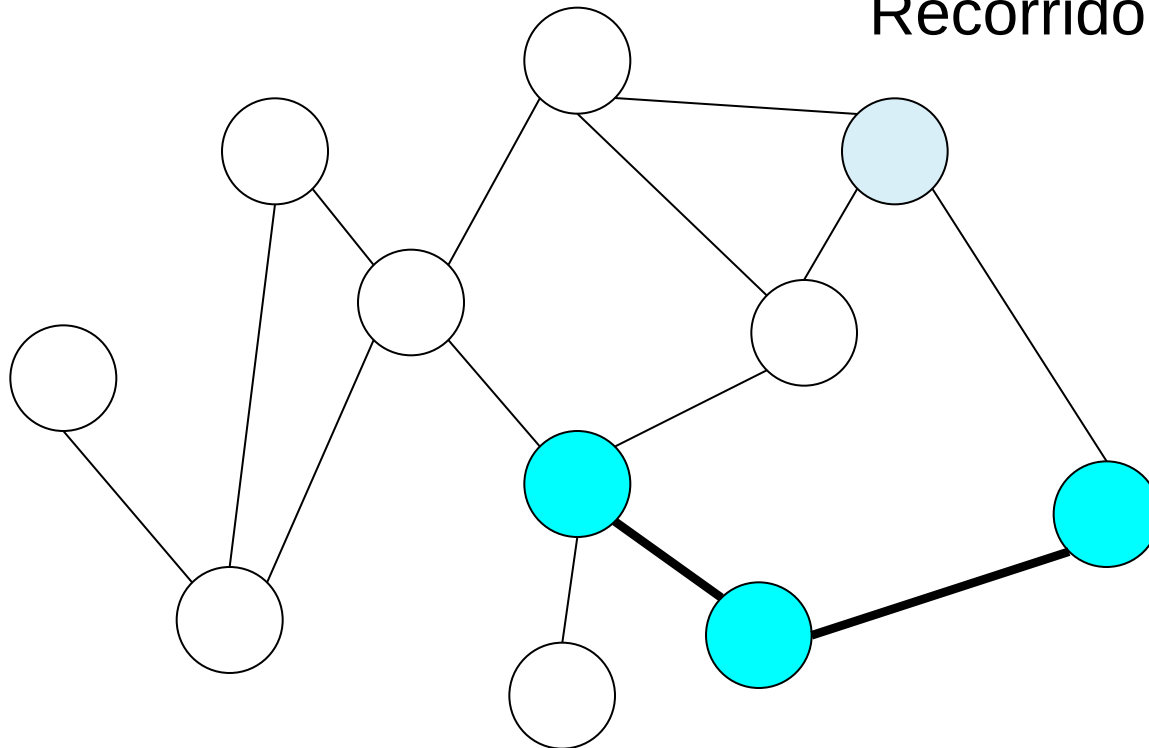
Recorrido en profundidad



Se elige vecino para avanzar

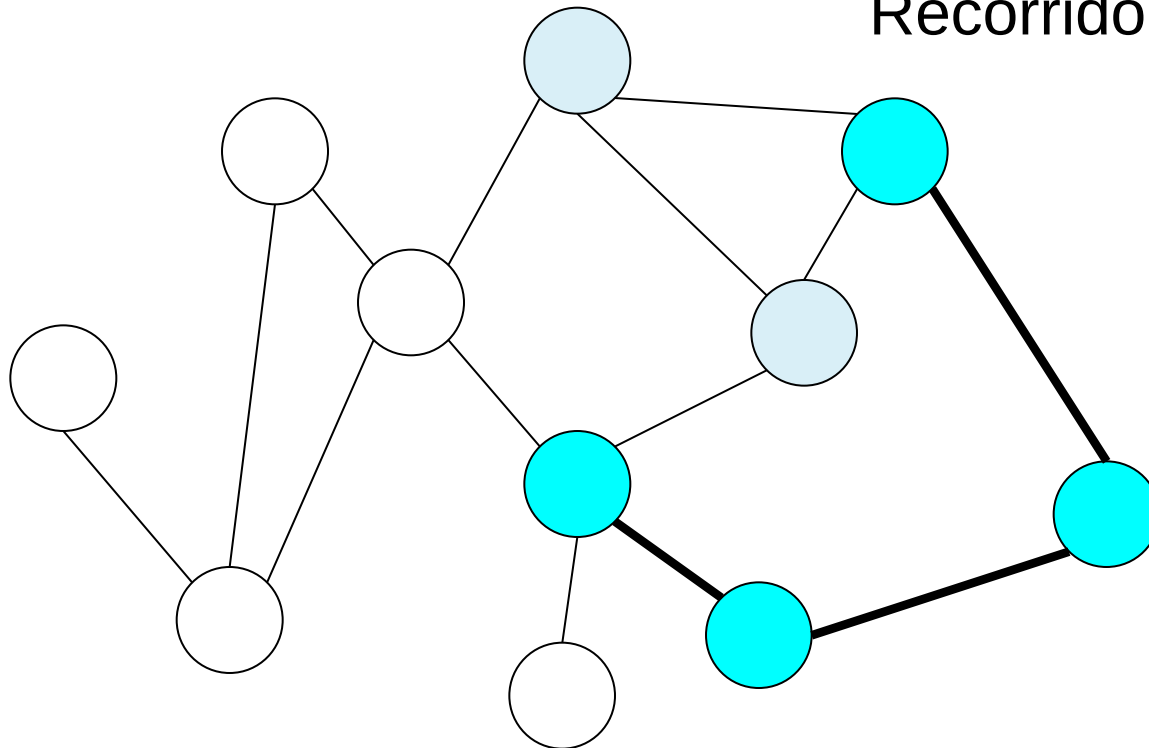
Recorridos

Recorrido en profundidad



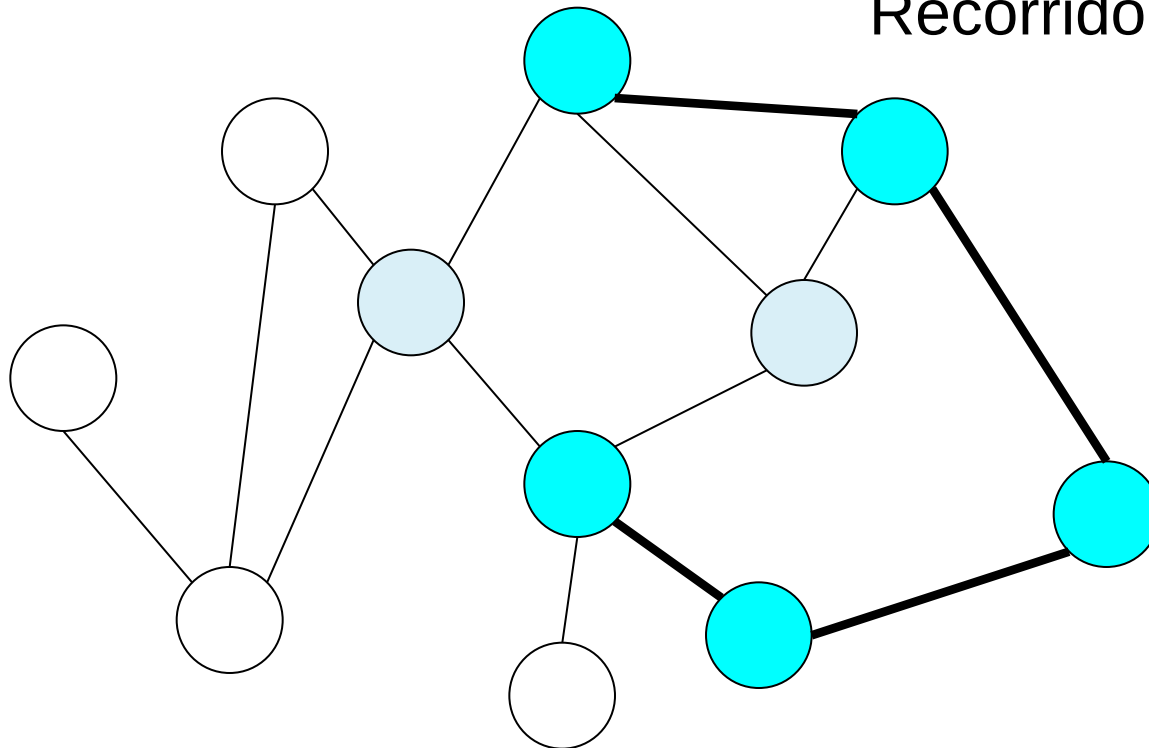
Recorridos

Recorrido en profundidad



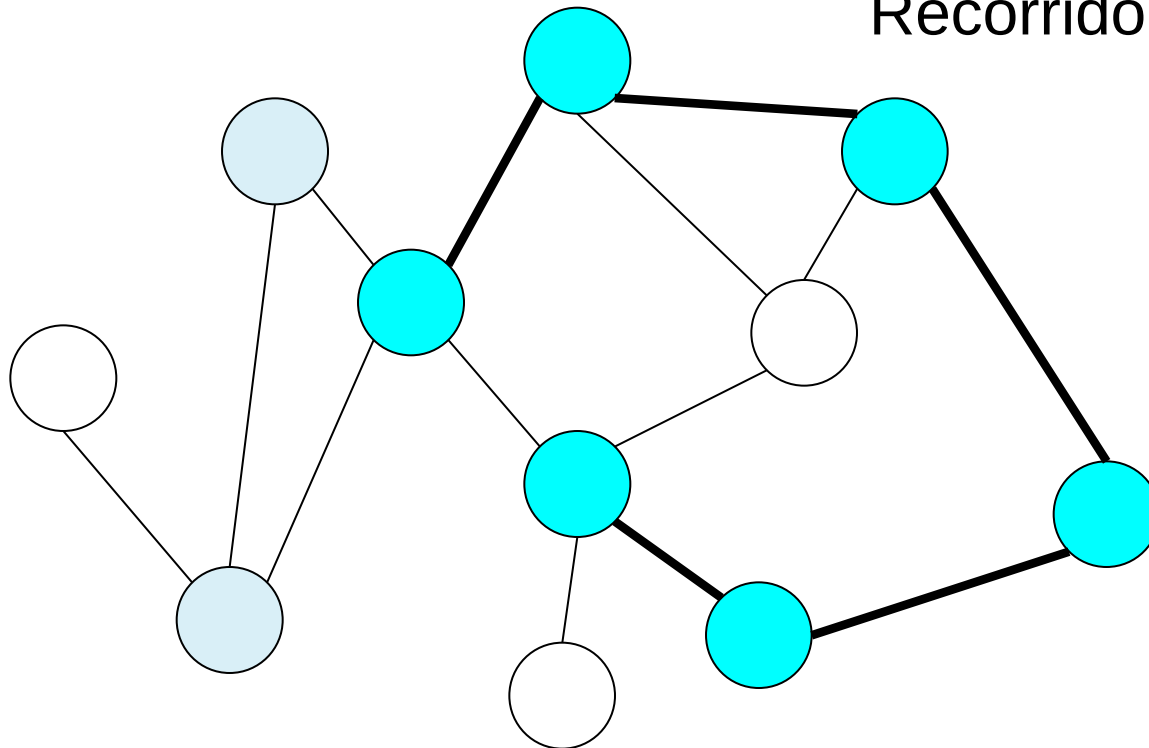
Recorridos

Recorrido en profundidad



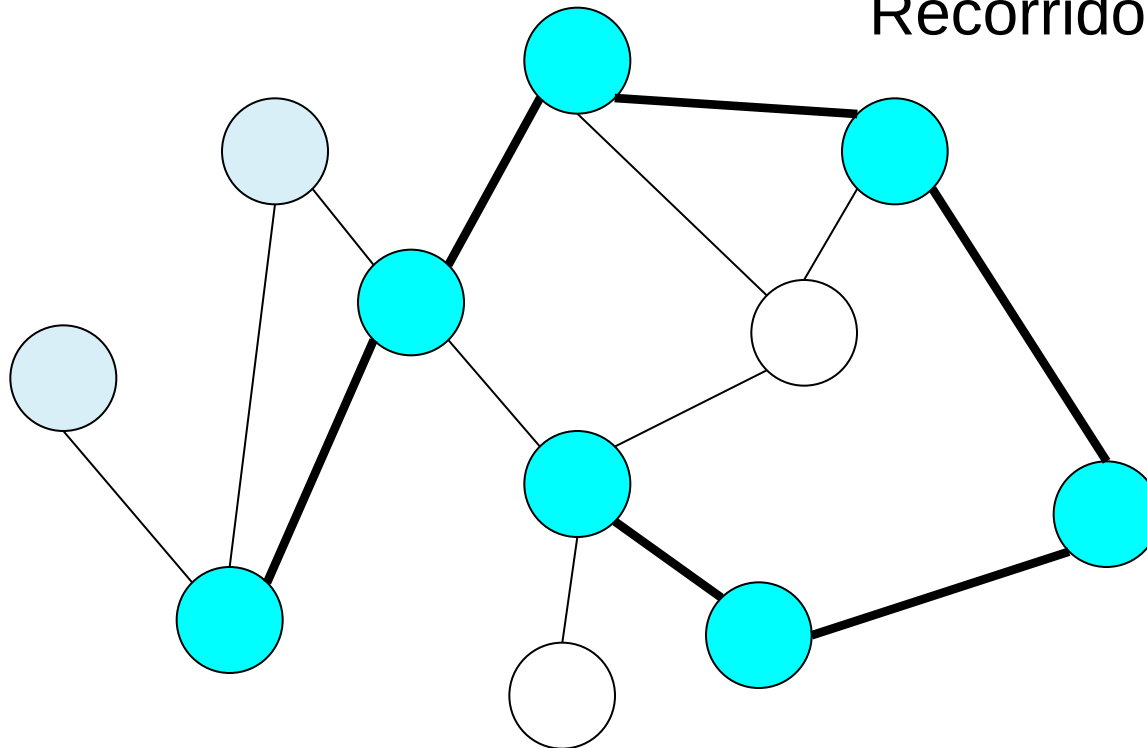
Recorridos

Recorrido en profundidad



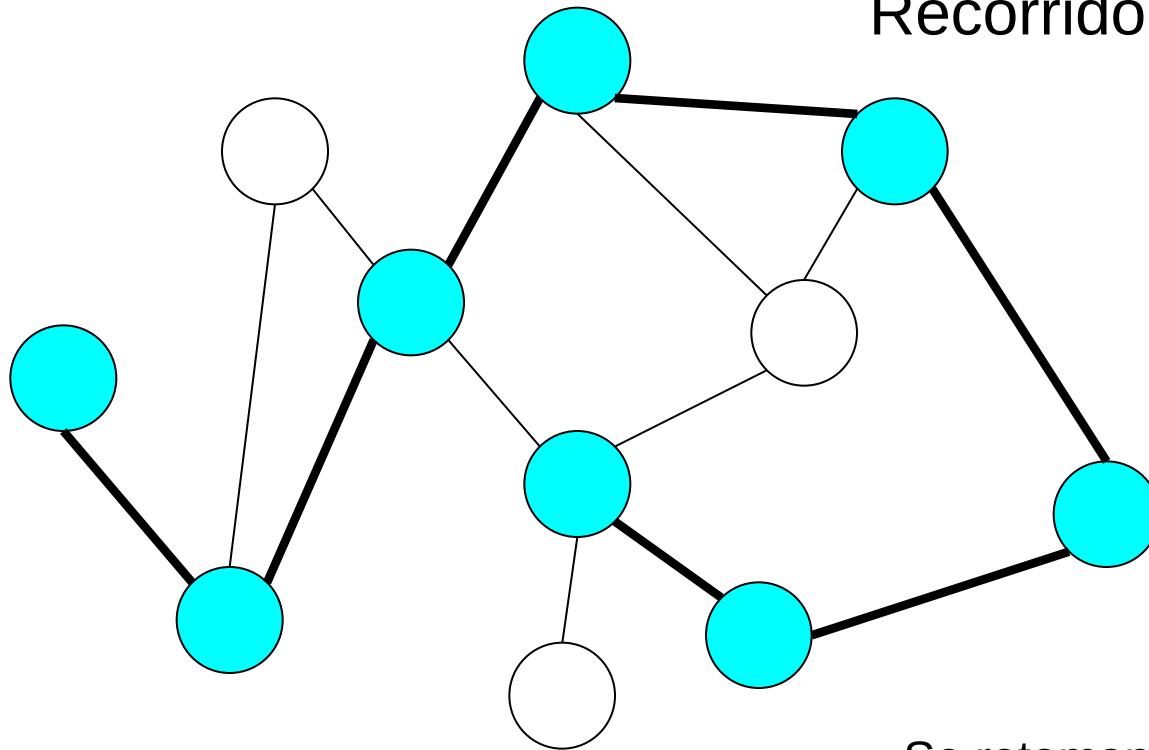
Recorridos

Recorrido en profundidad



Recorridos

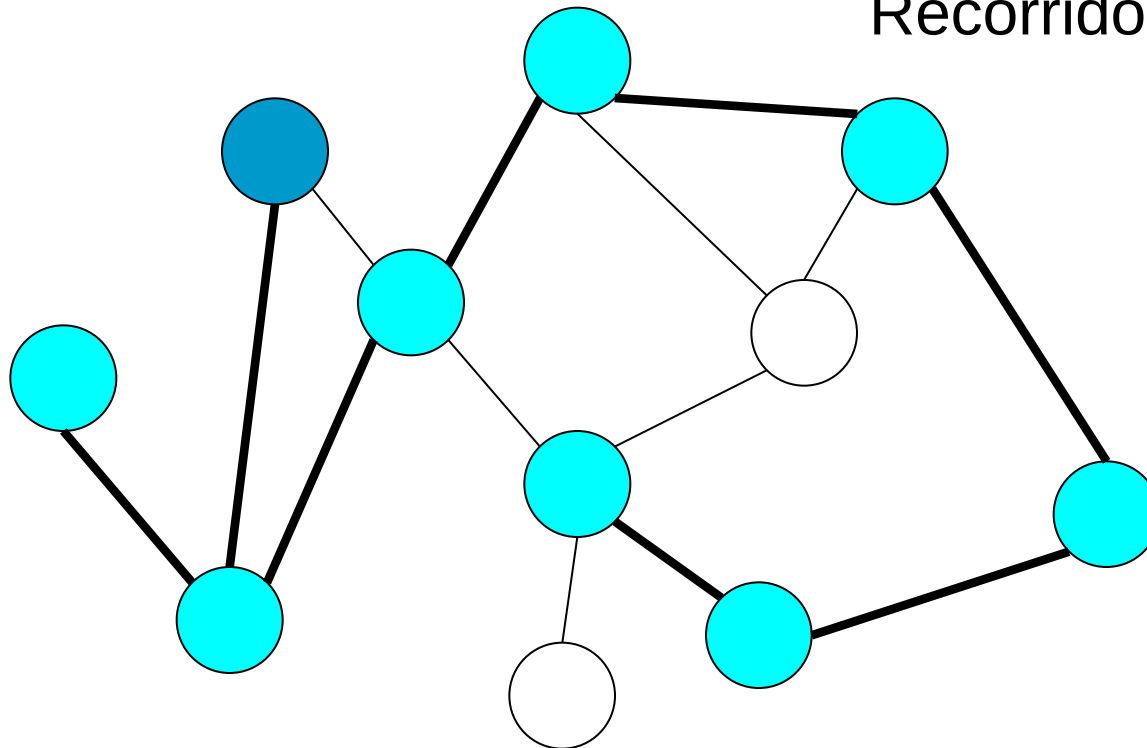
Recorrido en profundidad



Se retoman caminos no tomados

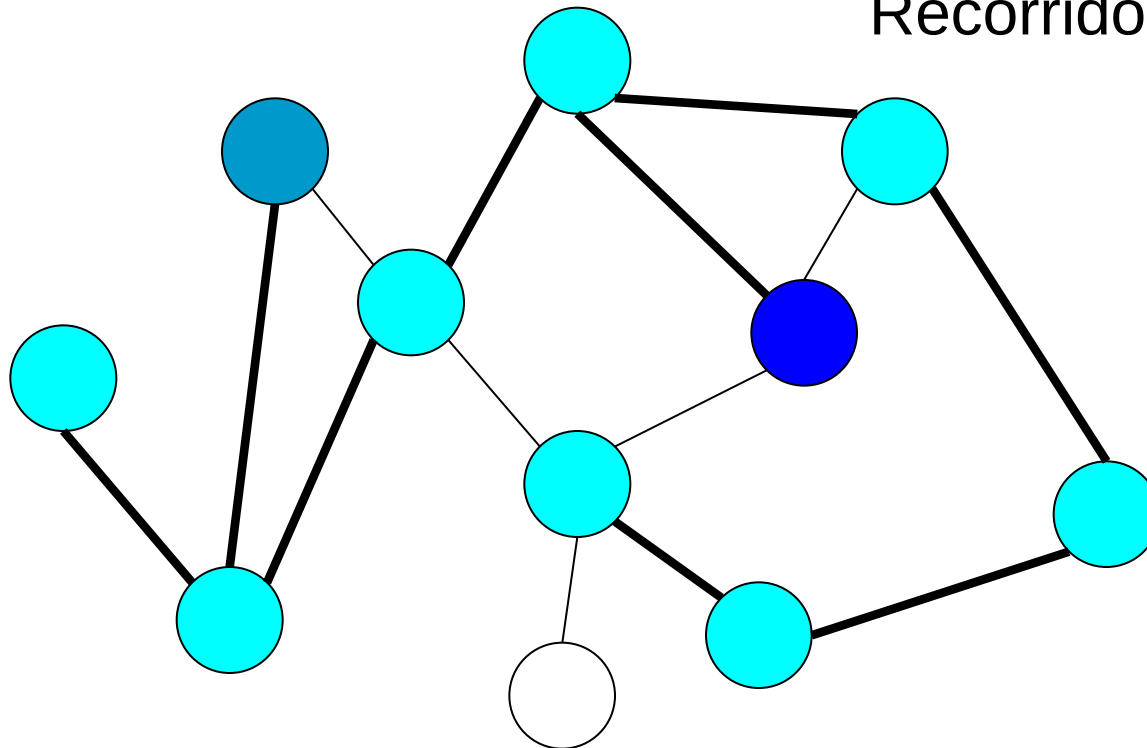
Recorridos

Recorrido en profundidad



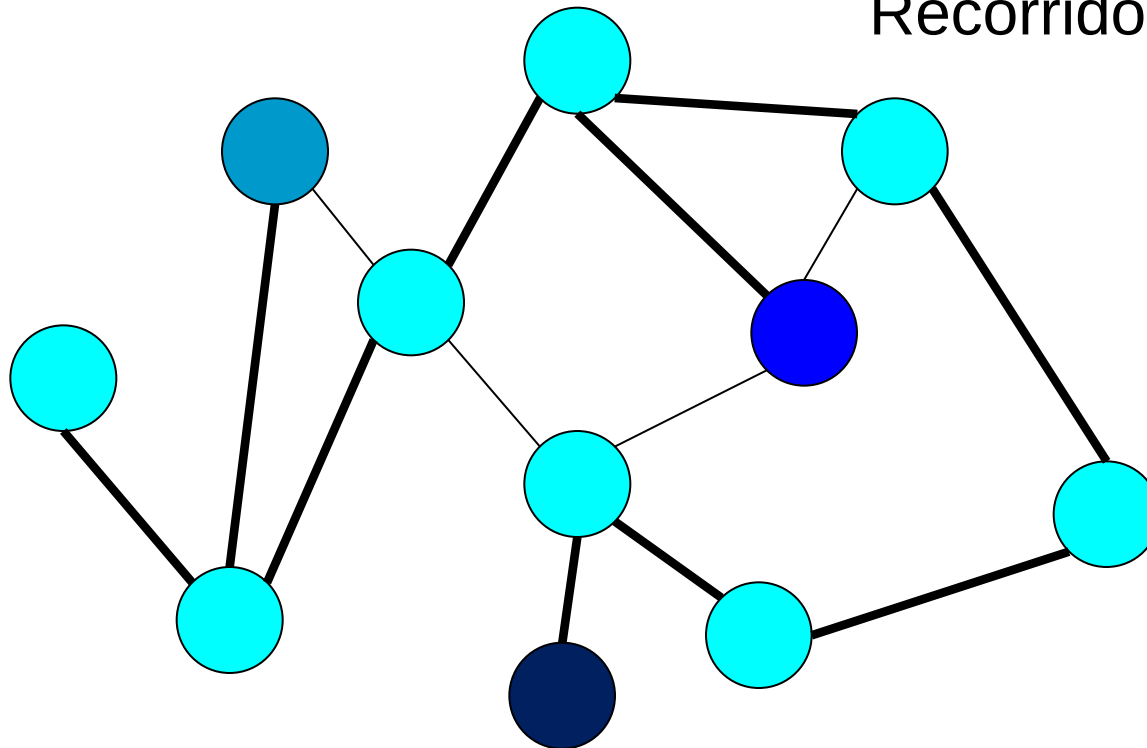
Recorridos

Recorrido en profundidad



Recorridos

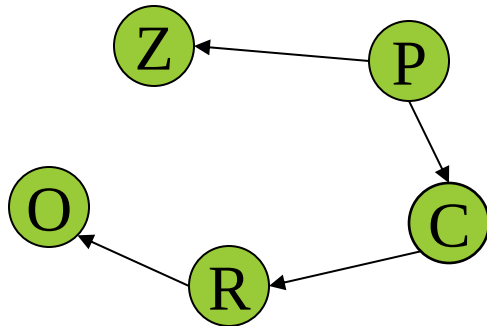
Recorrido en profundidad



Recorridos

Recorrer en anchura: se visita el nodo de partida, para después visitar los adyacentes no visitados aún

- Estructura auxiliar: Cola para guardar los adyacentes no visitados



Cola (queue)

	P	
	Z	C
	C	
	R	
	O	

Conjunto visitados

{P}

{P,Z}

{P,Z,C}

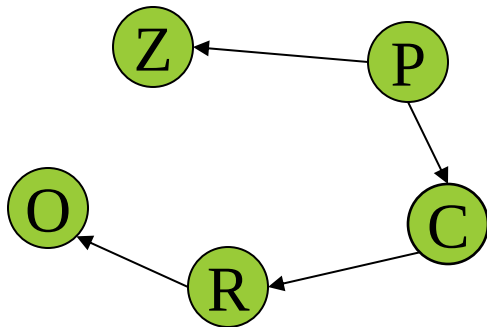
{P,Z,C,R}

{P,Z,C,R,O}

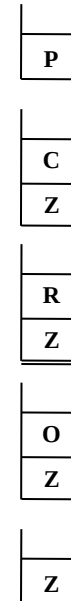
Recorridos

Recorrer en profundidad: se visita el nodo de partida, para después visitar en profundidad los adyacentes no visitados aún

- Estructura auxiliar: Pila para guardar los adyacentes no visitados



Pila (stack)



{P} Conjunto visitados

{P,C}

{P,C,R}

{P,C,R,O}

{P,C,R,O,Z}