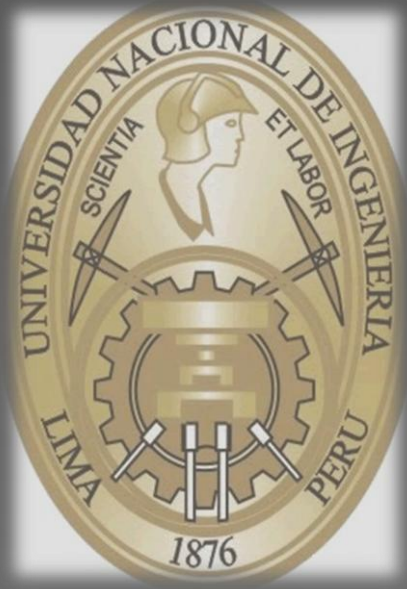


Universidad Nacional de Ingeniería
Facultad de Ciencias
Estructura de datos



BUSQUEDA EN GRAFOS

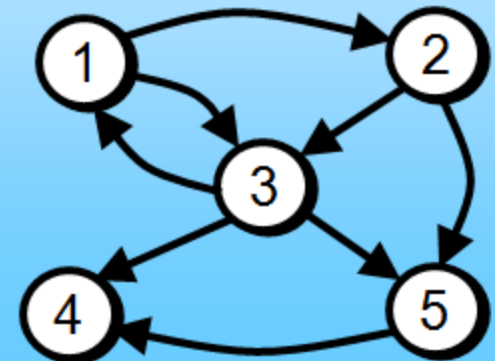
EXPOSITOR : Tomas J. Casas Rodriguez

2012

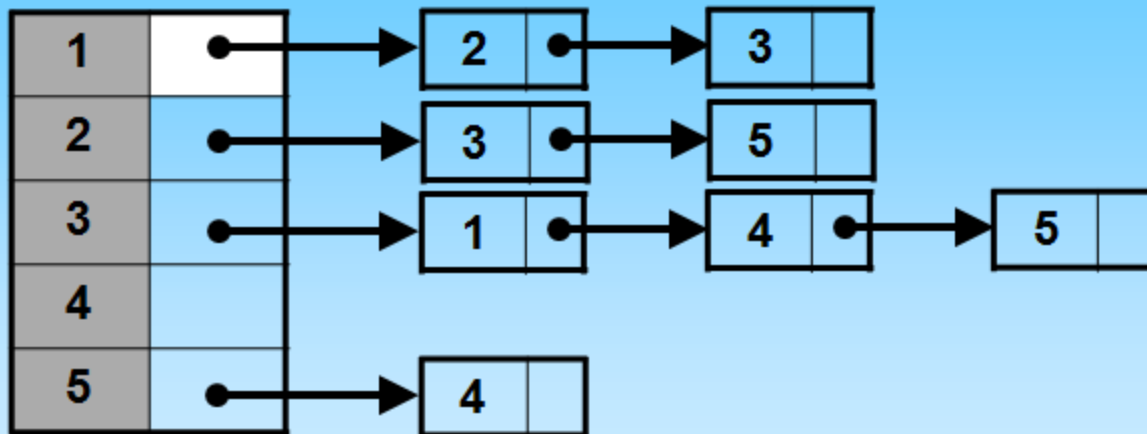
Representación de grafos.

- **Representación del conjunto de aristas, A.**
 - Mediante **matrices de adyacencia**.

| M | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | T | T | | |
| 2 | | | T | | T |
| 3 | T | | | T | T |
| 4 | | | | | |
| 5 | | | | T | |



- Mediante **listas de adyacencia**.



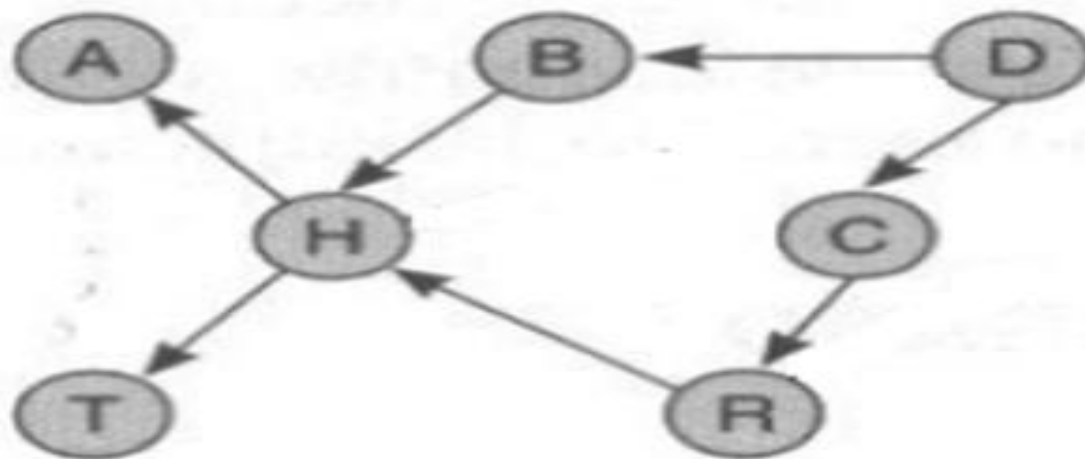
Recorridos sobre grafos.

- Idea similar al recorrido en un árbol.
- Se parte de un nodo dado y se visitan los vértices del grafo de manera ordenada y sistemática, *moviéndose* por las aristas.
- **Tipos de recorridos:**
 - **Búsqueda primero en profundidad.** Equivalente a un recorrido en preorden de un árbol.
 - **Búsqueda primero en amplitud o anchura.** Equivalente a recorrer un árbol por niveles.
- Los recorridos son una **herramienta** útil para resolver muchos problemas sobre grafos.

Búsqueda primero en profundidad.

- El recorrido **no es único**: depende del nodo inicial y del orden de visita de los adyacentes.
- El orden de visita de unos nodos a partir de otros puede ser visto como un árbol: **árbol de expansión en profundidad asociado al grafo.**
- Si aparecen varios árboles: **bosque de expansión en profundidad.**

El recorrido empieza por un vértice v del grafo; éste se marca como visitado y se *mete* en la pila. Después se recorre en profundidad cada vértice adyacente a v no visitado; así hasta que no haya más vértices adyacentes no visitados. Esta estrategia de examinar los nodos se denomina en profundidad porque la dirección de «visitar» es hacia *adelante* mientras que sea posible; al contrario que la búsqueda en anchura que primero visita todos los vértices posibles en *amplitud*.



Pila

D

B C

B R

B H

B A T

B A

B

pila vacía

Vértices procesados

D

C

R

H

T

A

B

Implementacion en c++

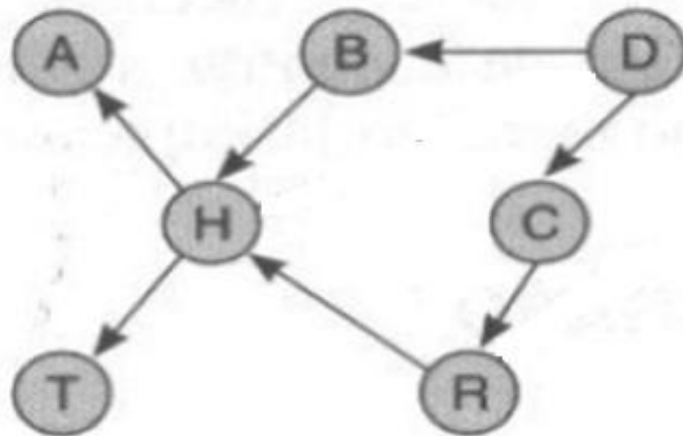
```
int profundidad(grafo g, ppila p, char clave[])
{
    push(p, g, g->x);
    while(p!=NULL)
    {
        imprimir(p);
        g=pop(p);
        cout<<"\n\t\tVertice visitado : "<<g->x<<endl;
        g->visto=1;
        Lista a=g->lista;
        while(a!=NULL){
            if(a->destino->visto==0){
                if(strcmp(a->destino->x, clave)==0)
                    return 1;
                push(p, a->destino, a->destino->x);
            }
            a=a->enlace;
        }
    }
    cout<<endl;
    return 0;
}
```

Búsqueda primero en anchura (o amplitud).

- **Búsqueda en anchura** empezando en un nodo v :
 - Primero se visita v .
 - Luego se visitan todos sus adyacentes.
 - Luego los adyacentes de estos y así sucesivamente.
- El algoritmo utiliza una **cola de vértices**.
- Operaciones básicas:
 - Sacar un elemento de la cola.
 - Añadir a la cola sus adyacentes no visitados.

El orden de procesar o visitar los nodos, en el recorrido en anchura, se puede expresar de manera más concisa en estos pasos:

1. *Marcar* el vértice de partida v .
2. *Meter* en la cola el vértice de partida v .
3. Repetir los pasos 4 y 5 hasta que se cumpla la condición *cola vacía*.
4. *Quitar* nodo frente de la cola, w , visitar w .
5. *Meter* en la cola todos los vértices adyacentes a w que no estén marcados, a continuación marcar esos vértices.
6. Fin del recorrido.



Cola

D

B C

C H

H R

R A T

A T

T

cola vacía

Vértices procesados

D

B

C

H

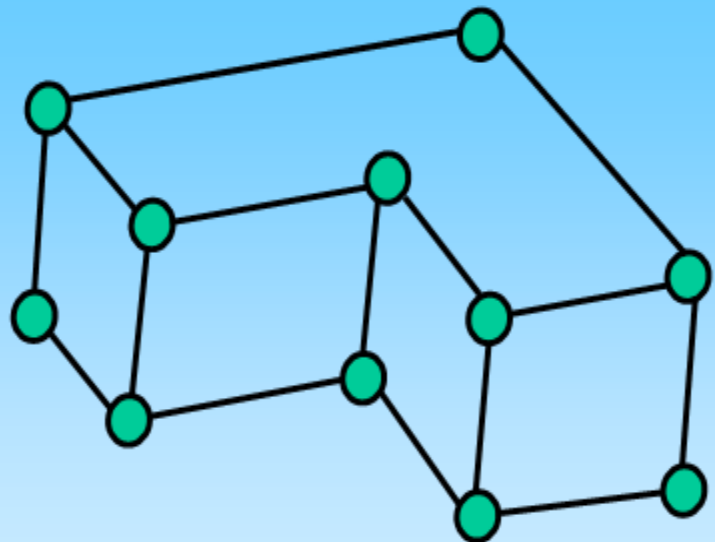
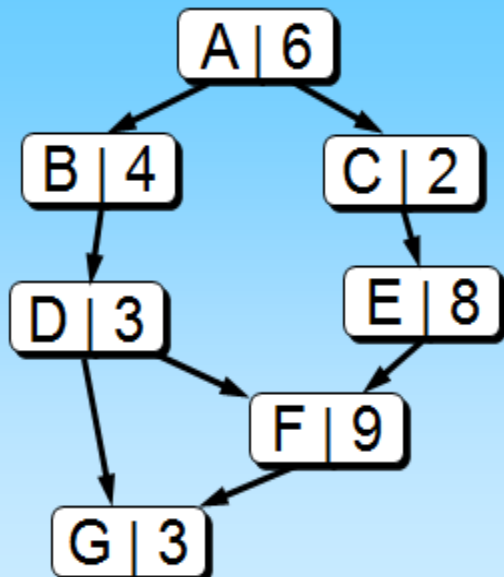
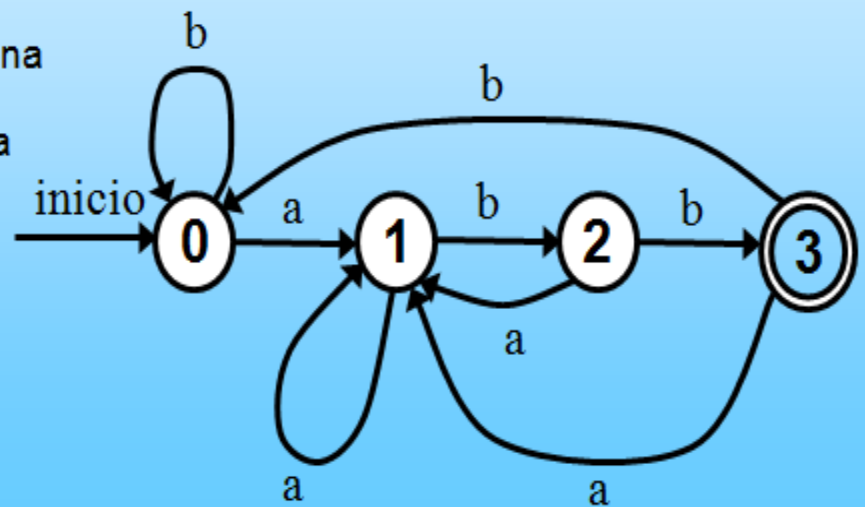
R

A

T

Implementacion en c++

```
int anchura(grafo g,kola &q,char clave[])
{
    encola(q,g,g->x);
    while(q.delante!=NULL)
    { vercola(q);
      g=desencolar(q);
      cout<<"\n\t\tVertice visitado : "<<g->x<<endl;
      g->visto=1;
      Lista a=g->lista;
      while(a!=NULL){
          if(a->destino->visto==0){if(strcmp(a->destino->x,clave)==0)
                                  return 1;
                                  encola(q,a->destino,a->destino->x);
          }
          a=a->enlace;
      }
    }
    cout<<endl;
    return 0;
}
```



BIBLIOGRAFIA

->Luis Joyanes Aguilar," Algoritmos y estructura de datos,una perspectiva en C", 1ra. Edición, Edit. Mc Graw Hill, 2004(517-522pages)

GRACIAS