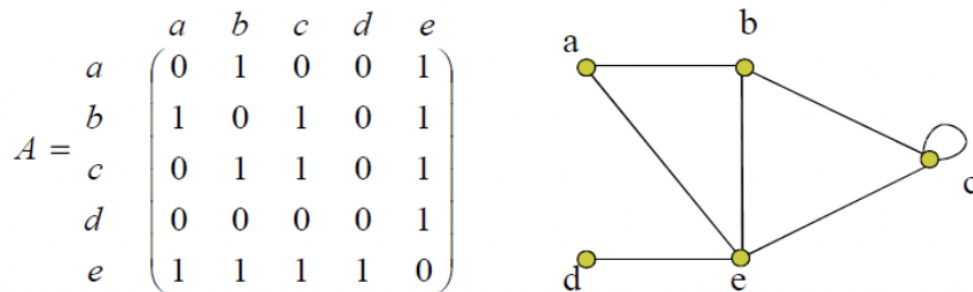


Caso de estudio: Matriz de adyacencia con lenguaje C

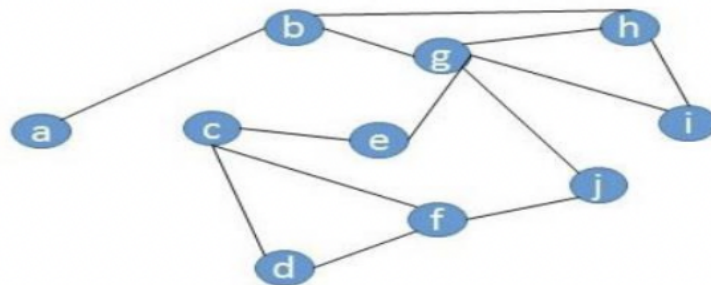
En la escuela de Telecomunicaciones, a Luis Dávila le están enseñando la manera de conectar diferentes equipos de cómputo para transferir diferentes paquetes de información y esto lo puede representar de manera fácil mediante el manejo de gráficas, mejor conocidos como grafos.

Uno de los elementos importantes que Luis debe dominar y tener en cuenta es la representación de las rutas mediante matrices. A continuación, se muestra un ejemplo de un grafo y una matriz de adyacencia (DMD5, 2013):



Tomando como referencia este caso, observamos que, si se suman las columnas y/o renglones, se puede obtener el grado de un vértice.

Como primera etapa del proyecto que Luis quiere desarrollar y presentárselo a su jefe realizó un programa en lenguaje C, donde el puede ingresar el número de nodos que desee y represente la matriz de adyacencia. A continuación, se muestra un ejemplo de un grafo:



Luis realizó el siguiente programa usando la estructura de arreglos en lenguaje C y así pudo representar y llenar la matriz de adyacencia del ejemplo:

```

#include <stdio.h>
#include <stdio.h>#
int main(int argc, char *argv[])
{
int tam, contador, contador2, contador3;
char a=97; //es el número del código ASCII de la letra "a" y ahí comienzan las
minúsculas
printf("Dame el numero de nodos: ");// se designa por el usuario
scanf("%d",&tam);
int arreglo[tam][tam]; //Se define el tamaño del arreglo proporcionado por el
usuario
printf(" ");
for (contador3=0; contador3<tam; contador3++ ) {
printf("%c",a++);
} //En este ciclo que termina simplemente se nombran las columnas y
renglones con las letras de la a hasta el
printf("\n"); //caracter que le corresponda
a=97; // Se reinicia la a la letra a
for(contador2=0; contador2<tam; contador2++){
printf("%c",a++);
for (contador=0; contador<tam; contador++){

arreglo[contador][contador2]=0; //En esta parte se llena la matriz con ceros
printf("%d",arreglo[contador][contador2]);
} // fin del for
printf("\n");
} // fin del for del contador2
// En la siguiente parte del programa, se pondrá un 1 en donde haya
char x, y;
int nx, ny;
char resp='s';
do{//Se repite hasta que no se quiera introducir un nodo más
fflush(stdin); // Se limpia el búfer de entrada
printf("Ingresa la letra (x): ");//Se refiere al eje x la letra correspondiente
scanf("%c",&x);
printf("Ingresa la letra (y): ");// La del eje Y
fflush(stdin);
scanf("%c",&y);
nx=x-97; //Se resta para obtener la posición del registro y se convierte
ny=y-97; // en una variable numérica
arreglo[nx][ny]=1; // Se asigna el uno a la posición que le corresponde
// SE MUESTRA NUEVAMENTE EL ARREGLO
a=97;
printf(" ");
for (contador3=0; contador3<tam; contador3++ ) {

```

```

printf("%c",a++);
}
printf("\n");
a=97;
for(contador2=0; contador2<tam; contador2++){
printf("%c",a++);
a=97;
for(contador2=0; contador2<tam; contador2++){
printf("%c",a++);
for (contador=0; contador<tam; contador++){
//arreglo[contador][contador2]=0;
printf("%d",arreglo[contador][contador2]);
// fin del for
printf("\n");
} // fin del for del contador2
printf("Deseas ingresar otra interseccion? s/n\n: ");
fflush(stdin);
scanf("%c",&resp);
}while(resp!='n' && resp!='N'); // Al responder que no, se termina el programa
system("PAUSE");
return 0;
}

```

Se presenta a continuación la salida del programa, donde Luis puede ir añadiendo los lazos que unen a los nodos. Como habrás podido notar, el programa tiene un ciclo que termina hasta que el usuario desee dejar de ingresar datos:

```

Dame el numero de nodos: 10
 abcdefghij
a00000000000
b00000000000
c00000000000
d00000000000
e00000000000
f00000000000
g00000000000
h00000000000
i00000000000
j00000000000
Ingresa la letra (x): a
Ingresa la letra (y): b
 abcdefghij
a00000000000
b10000000000
c00000000000
d00000000000
e00000000000
f00000000000
g00000000000
h00000000000
i00000000000
j00000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): b
Ingresa la letra (y): g
 abcdefghij
a00000000000
b10000000000
c00000000000
d00000000000
e00000000000
f00000000000
g01000000000
h00000000000
i00000000000
j00000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): b
Ingresa la letra (y): h
 abcdefghij
a00000000000
b10000000000
c00000000000
d00000000000
e00000000000
f00000000000
g01000000000
h01000000000
i00000000000
j00000000000
Deseas ingresar otra interseccion? s/n

```

```

: s
Ingresa la letra (x): c
Ingresa la letra (y): f
  abcdefghij
a0000000000
b1000000000
c0000000000
d0000000000
e0000000000
f0010000000
g0100000000
h0100000000
i0000000000
j0000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): c
Ingresa la letra (y): d
  abcdefghij
a0000000000
b1000000000
c0000000000
d0010000000
e0000000000
f0010000000
g0100000000
h0100000000
i0000000000
j0000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): g
Ingresa la letra (y): e
  abcdefghij
a0000000000
b1000000000
c0000000000
d0010000000
e0000001000
f0010000000
g0100000000
h0100000000
i0000000000
j0000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): g
Ingresa la letra (y): h
  abcdefghij
a0000000000
b1000000000
c0000000000
d0010000000
e0000001000
f0010000000
g0100000000
h0100001000
i0000000000
j0000000000
Deseas ingresar otra interseccion? s/n

```

```

Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): g
Ingresa la letra (y): i
  abcdefghij
a00000000000
b10000000000
c00000000000
d00100000000
e00000010000
f00100000000
g01000000000
h01000010000
i00000010000
j00000000000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): g
Ingresa la letra (y): j
  abcdefghij
a00000000000
b10000000000
c00000000000
d00100000000
e00000010000
f00100000000
g01000000000
h01000010000
i00000010000
j00000010000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): f
Ingresa la letra (y): c
  abcdefghij
a00000000000
b10000000000
c00000100000
d00100000000
e00000010000
f00100000000
g01000000000
h01000010000
i00000010000
j00000010000
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): d
Ingresa la letra (y): f
  abcdefghij
a00000000000
b10000000000
c00000100000
d00100000000
e00000010000
f00110000000
g01000000000
h01000010000
i00000010000
j00000010000

```

```
Deseas ingresar otra interseccion? s/n
: s
Ingresa la letra (x): f
Ingresa la letra (y): j
 abcdefghij
a00000000000
b10000000000
c0000010000
d00100000000
e00000010000
f00110000000
g01000000000
h01000010000
i00000010000
j00000110000
Deseas ingresar otra interseccion? s/n
: n
Presione una tecla para continuar . . . _
```

Referencias

- DSpace en ESPOL. (2013). DMDM5 Capítulo 5: Teoría de gráficas. En Matemáticas discretas. Recuperado: febrero 09, 2020. Sitio web: http://www.dspace.espol.edu.ec/retrieve/2767/DMD5_teoría_de_gráficas.pdf