

Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas
Alumna: **Mayra Alejandra Rivera Rodríguez**
Matricula: **1742899**

Matemáticas Computacionales

Reporte de Fila, Pila, Grafo, BFS y DFS

En este reporte se expondrá de manera breve la explicación de que es una fila, una pila, un grafo, el BFS y el DFS, además se mostrara un código de cada uno de ellos y se explicara la forma en la que funciona.

1-Fila

Es una estructura de datos la cual nos permite juntar elementos y de ellos obtener el primero, además de esto también nos permite conocer la cantidad de elementos que hay en la estructura.

El siguiente código es un ejemplo de una fila en Python

class Fila:

```
    def __init__(self):  
        self.fila=[]  
    def obt(self):  
        return self.fila.pop()  
    def met(self,e):  
        self.fila.insert(0,e)  
    @property  
    def lon (self):  
        return len(self.fila)
```

El código anterior lo que hace es que te permite ingresar elementos por medio de la función met, además la función obt es para cumplir la función principal del código que es arrojar el primer elemento de la fila y por último la función lon nos indica la cantidad de elementos ingresados o que contiene la fila.

2.-Pila

Es una estructura de datos la cual nos permite juntar elementos y de ellos obtener el ultimo, además de esto también nos permite conocer la cantidad de elementos que hay en la estructura.

El siguiente código es un ejemplo de una pila en Python

```
class Pila:
    def __init__(self):
        self.pila=[]
    def obt (self):
        return self.pila.pop()
    def met (self,e):
        self.pila.append(e)
    @property
    def lon (self):
        return len (self.pila)
```

El código anterior lo que hace es que te permite ingresar elementos por medio de la función met, además la función obt es para cumplir la función principal del código que es arrojar el último elemento de la fila y por último la función lon nos indica la cantidad de elementos ingresados o que contiene la pila.

3.- Grafo

Se puede definir un grafo como un conjunto de vértices que también son llamados nodos que se encuentran conectados por medio de aristas. Estas estructuras sirven mas que nada para representar de manera

general situaciones de la vida cotidiana como relaciones de amistad, relaciones entre caminos , entre otras cosas.

El siguiente código es un ejemplo de una grafo en Python

class Grafo:

```
def __init__(self):
    self.M = set()
    self.R = dict()
    self.vecinos = dict()

def agregar(self, m):
    self.M.add(m)
    if not m in self.vecinos:
        self.vecinos[m] = set()

def conect(self, m, a, peso=1):
    self.agregar(m)
    self.agregar(a)
    self.R[(m, a)] = self.R[(a, m)] = peso
    self.vecinos[m].add(a)
    self.vecinos[a].add(m)
```

el código lo que presenta como vecinos es cada uno de los vértices y por medio de la función conect los va uniendo por medio de aristas para formar un grafo en el cual iremos visitando a cada uno de los vértices en cierto orden de manera que solo pase una vez por cada uno de ellos.

4.- BFS

Es un algoritmo el cual sus siglas se refieren a Breadth First Search que en español se podría interpretar como búsqueda en amplitud. Este algoritmo toma como base un vértice inicial y realiza un recorrido por

todo el grafo visitando primero a los vértices mas cercanos como si fuera una onda de expansión de un radar o algo asi.

El siguiente código es un ejemplo del algoritmo BFS en Python

```
def B_F_S(g,ni):
    visit=[]
    f=Fila()
    f.met(ni)
    while(f.lon>0):
        na=f.obt()
        visit.append(na)
        ra=g.vecinos[na]
        for nodo in ra:
            if nodo not in visit:
                f.met(nodo)
    return visit
```

Lo que nos permite este código es decirle cual será nuestro vértice inicial para que de ahí tome como base para continuar analizando el grafo primero con los vértices mas cercanos hasta el más lejano y lo que nos imprime en la forma en la que fueron visitados todos y cada uno de los vértices.

5.- DFS

Es un algoritmo es cual sus siglas se refieren a Depth First Search que en español se podría interpretar como Búsqueda en profundidad. Este algoritmo toma como base un vértice inicial y realizará un recorrido por el grafo visitando a los vértices que se encuentren de manera adyacente como si nos desplazáramos caminando sobre ellos.

El siguiente código es un ejemplo del algoritmo DFS en Python

```
def D_F_S(g,ni):
    visit=[]
```

```
f=Pila()
f.met(ni)
while(f.lon>0):
    na=f.obt()
    visit.append(na)
    ra=g.vecinos[na]
    for nodo in ra:
        if nodo not in visit:
            f.met(nodo)
return visit
```

Lo que nos permite este código es decirle cual será nuestro vértice inicial para que de ahí tome como base para continuar analizando el grafo primero con los vértices mas cercanos de manera que estos sean adyacentes al anterior y así continua hasta el ultimo y lo que nos imprime en la forma en la que fueron visitados todos y cada uno de los vértices.