

Public class PARCIAL {

Todos Caminos PASANDO A LISTA DE CIUDADES

Public LISTA Generica<String>

Resolve (Grafo<String> CIUDADES,

String Origen, String Destino, List<String> PASANDO POR)

-- LISTA Generica<String> CAMINO = new LISTA Enlazada Generica();

-- IF (!CIUDADES.esVacio()) {

INT POS = BUSCAR_OYD (CIUDADES.LISTA DE VERTICES(), Origen, Destino)

IF (POS != -1) {

Boolean[] MARCA = new Boolean[CIUDADES.LISTA DE VERTICES().TAMANO()+1]();

INT[] V-PASANDO POR = new INT[CIUDADES.LISTA DE VERTICES().TAMANO()+1]();

MARCA PASANDO POR (V-PASANDO POR, PASANDO POR, CIUDADES.LISTA DE VERTICES());

DFS(POS, CIUDADES, Destino, MARCA, V-PASANDO POR, CAMINO, PASANDO POR.TAMANO())

}

RETURN CAMINO;

}

Private INT BUSCAR_OYD (LISTA Generica<VERTICES<String>> LISTA, String Or, String Dest,)

INT = Origen = -1, Destino = -1;

LISTA = COMENZAR();

While ((Origen == -1 || Destino == -1) && (!LISTA.FIN())) {

VERTICE<String> V = LISTA.PROXIMO();

IF ((Origen == -1) && (V.DATO().equals(Or))) {

Origen = V.POSICION();

} else IF ((Destino == -1) && (V.DATO().equals(Dest))) {

Destino = V.POSICION();

}

IF (Destino == -1) {

Origen = -1;

}

RETURN Origen;


```
PRIVATE VOID MARCARPASADO POR (INT[] MARCA, LISTAgenerica<string> CIUDADES,  
LISTAgenerica<vertice> <string> LISTA-V);
```

```
LISTA-V.comenzar();
```

```
WHILE(!LISTA-V.FIN()) {
```

```
vertice<string> V = LISTA-V.proximo();
```

```
IF (CIUDADES.incluye(V.DATO())) {
```

```
MARCA[V.POSICION()] = 1
```

```
}
```

```
}
```

```
}
```

```
PRIVATE Boolean DFS (INT POS, GRAFO<string> G, string DESTINO, Boolean[] MARCA,  
INT[] XVISITAR, LISTAgenerica<string> CAMINO, INT VISITADOS) {
```

```
Boolean cumple = false;
```

```
MARCA[POS] = true; Boolean cumple = false;
```

```
vertice<string> V = G.vertice(POS);
```

```
CAMINO.agregarFinal(V.DATO());
```

```
VISITADOS -= XVISITAR[V.POSICION()];
```

```
IF ((V.DATO.equals(DESTINO) && VISITADOS == 0) {
```

```
    cumple = true;
```

```
} else {
```

```
LISTAgenerica<ARISTA<string>> ADY = G.listaDeAdyacentes(V);
```

```
ADY.comenzar();
```

```
WHILE ((!cumple) && (!ADY.FIN())) {
```

```
ARISTA<string> A = ADY.proximo();
```

```
INT J = A.verticeDestino().POSICION();
```

```
IF (!MARCA[J]) {
```

```
    cumple = DFS(J, G, DESTINO, MARCA, XVISITAR, CAMINO, VISITADOS);
```

```
    }
```

```
}
```

```
MARCA[POS] = false;
```

```
CAMINO.eliminarEn(CAMINO.TAM-1);
```

```
RETURN cumple;
```