

Public class VisitarOsloUI

Public List GenerarCaminos (PaseoEnBici (Grafo<String> lugares, int maxTiempo, List GenerarCaminos lugares Restringidos)

List GenerarCaminos camino = new List Enlazado GenerarCaminos();

if (! lugares.esVacio()) {

int pos = buscar(lugares.listaDeVertices(), "ayuntamiento");

if (pos != -1) {

Boolean[] marcas = new Boolean [lugares.listaDeVertices().Tamaño];

marcarExcepciones (marcas, lugares Restringidos);

DFS (pos, lugares, camino, new List Enlazado GenerarCaminos(), marcas, maxTiempo);

}

}

return camino;

}

Private void DFS (int pos, Grafo<String> g, List GenerarCaminos canFinal, List GenerarCaminos canActual, Boolean[] marcas, int tiempo)

marcas[pos] = true

verice<String> vActual = g.verice(pos);

canActual.agregarFinal(vActual.dato());

List GenerarCaminos arista<String> aristas = g.listaDeVertices(vActual);

aristas.converger();

while (! aristas.Fin()) {

arista<String> arista = aristas.proximo();

int posSiguiente = arista.vericeDestino().posicion();

if (tiempo - arista peso() > 0) && (! marcas[posSiguiente]) {

DFS (posSiguiente, g, canFinal, canActual, marcas, tiempo - arista.peso());

}

}

if (canActual.tamano() > canFinal.tamano()) {

canFinal.converger();

while (! canFinal.Fin()) {

canFinal.eliminar(canFinal.proximo());

}

canActual.converger();

while (! canActual.Fin()) {

canFinal.agregarFinal(canActual.proximo());

}

}

marcas[pos] = false

canActual.eliminar(canActual.tamano());

}


```

private void marcarExcepciones (boolean[] marca, List<Gerente> gerentes, List<Gerente> gerentesExcepciones) {
    gerentesExcepciones.clear();
    while (!gerentes.isEmpty()) {
        int pos = buscar(gerentes, gerentesExcepciones);
        if (pos != -1) {
            marca[pos] = true;
        }
    }
}

```

```

private int buscar (List<Gerente> gerentes, String Buscado);
boolean excepciones = false;
int pos = -1;
gerentes.clear();
while (!excepciones && (!gerentes.isEmpty())) {
    Gerente gerente = gerentes.get(0);
    if (gerente.getId().equals(Buscado)) {
        excepciones = true;
        pos = gerente.getPosicion();
    }
}
return pos;

```