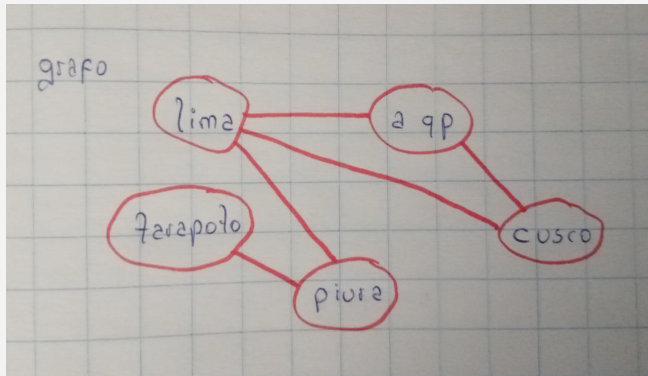


EJERCICIOS PROPUESTOS:

1. Enlace al repositorio GitHub:

<https://github.com/PedroSanMac/lab9-grafos.git>

2.



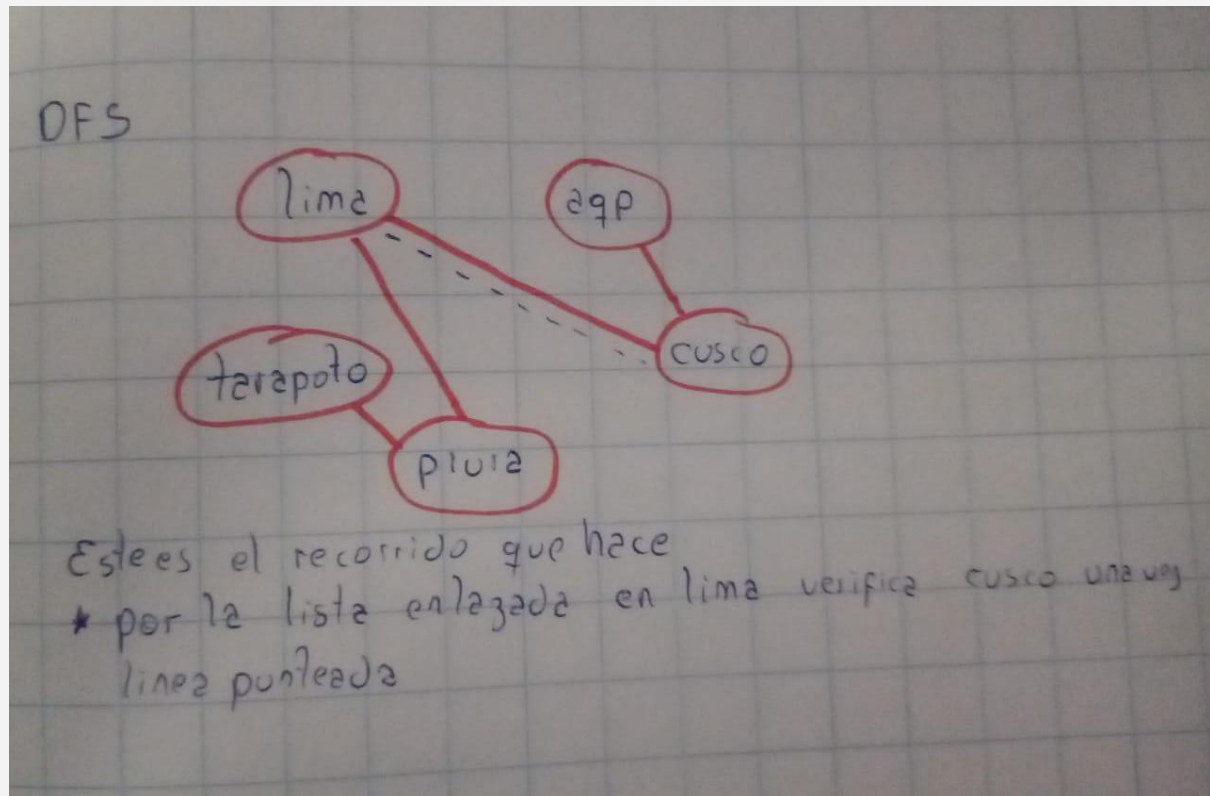
```
1
2 public class TestGrafo {
3
4     public static void main(String[] args) {
5
6         GrafoAdyacencia<String> g = new GrafoAdyacencia<String>();
7
8         g.insertVertice("lima");
9         g.insertVertice("aqp");
10        g.insertVertice("cusco");
11        g.insertVertice("piura");
12        g.insertVertice("tarapoto");
13
14        g.insertArista("lima", "aqp");
15        g.insertArista("cusco", "aqp");
16        g.insertArista("piura", "lima");
17        g.insertArista("cusco", "lima");
18        g.insertArista("piura", "tarapoto");
19
20        System.out.println("Grafo:\n" + g);
21
22    }
23 }
```

<terminated> TestGrafo [Java Application] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.justj.op

Grafo:
tarapoto --> piura,
piura --> tarapoto, lima,
cusco --> lima, aqp,
aqp --> cusco, lima,
lima --> cusco, piura, aqp, |

3.

DFS:



```
2 public class TestGrafo {
3
4 public static void main(String[] args) {
5
6     GrafoAdyacencia<String> g = new GrafoAdyacencia<String>();
7
8     g.insertVertice("lima");
9     g.insertVertice("aqp");
10    g.insertVertice("cusco");
11    g.insertVertice("piura");
12    g.insertVertice("tarapoto");
13
14    g.insertArista("lima", "aqp");
15    g.insertArista("cusco", "aqp");
16    g.insertArista("piura", "lima");
17    g.insertArista("cusco", "lima");
18    g.insertArista("piura", "tarapoto");
19
20    System.out.println("Grafo:\n" + g);
21
22    g.DFS("aqp");
23    System.out.println();
24 }
```

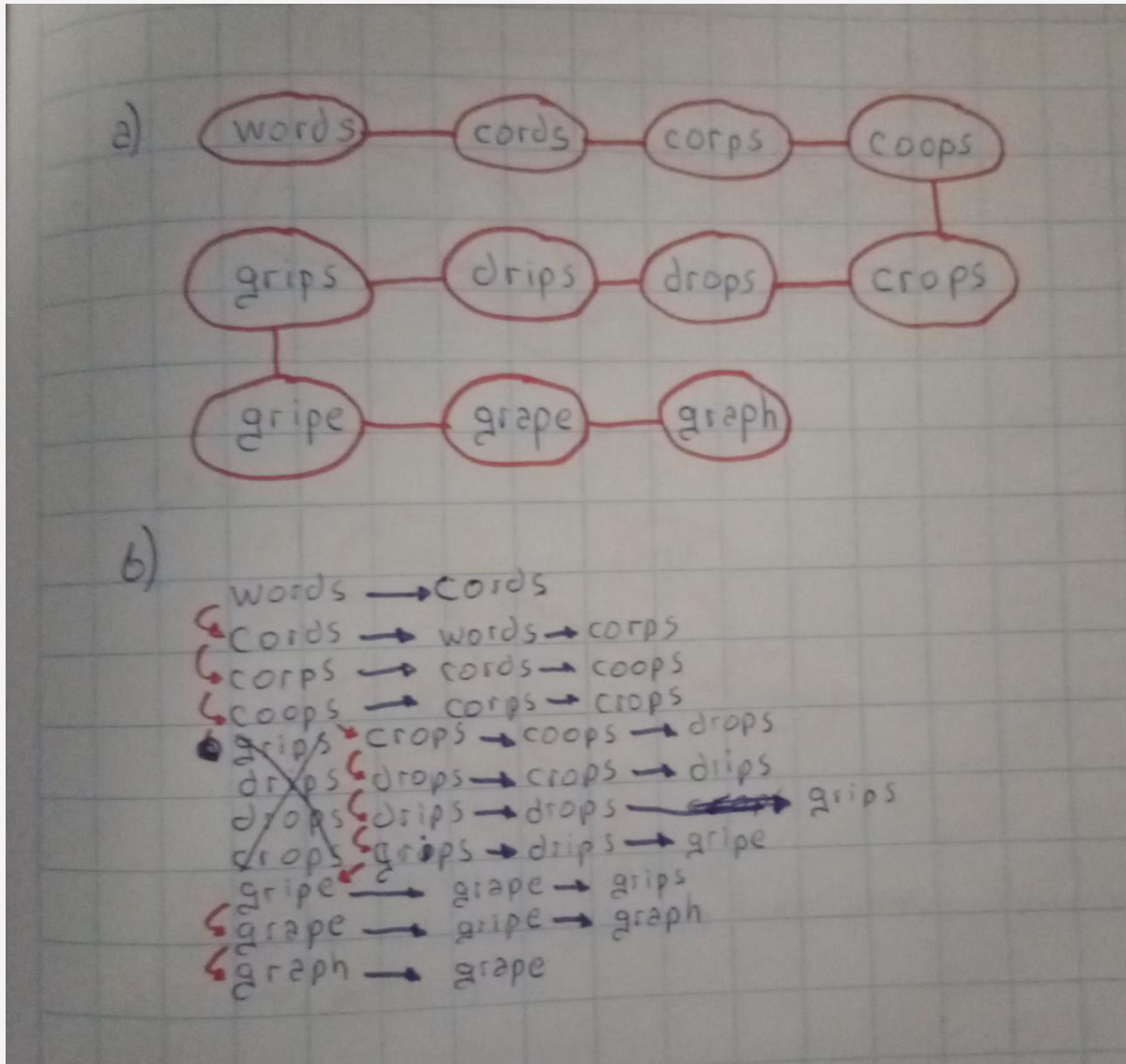
Problems @ Javadoc Declaration Console

<terminated> TestGrafo [Java Application] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justj.op

Grafo:
tarapoto --> piura,
piura --> tarapoto, lima,
cusco --> lima, aqp,
aqp --> cusco, lima,
lima --> cusco, piura, aqp, |
aqp, cusco, lima, piura, tarapoto,

4.

- a) Dibuje el grafo definido por las siguientes palabras: words cords corps coops crops drops drips grips gripe grape graph
- b) Mostrar la lista de adyacencia del grafo.



5.

CUESTIONARIO

1. ¿Cuántas variantes del algoritmo de Dijkstra hay y cuál es la diferencia entre ellas? (1 puntos)

El más famoso de los algoritmos para caminos de costo mínimo es Dijkstra. Pero es posible optimizar su rendimiento por eso existen muchas variantes de este algoritmo como por ejemplo:

Implementar a Dijkstra Montículos Binarios.

Implementar a Dijkstra Montículo de Fibonacci.

Algoritmo A*.

2. Investigue sobre los ALGORITMOS DE CAMINOS MÍNIMOS e indique, ¿Qué similitudes encuentra, qué diferencias, en qué casos utilizar y porque? (2 puntos)

	Similitud	Diferencias	Casos
Dijkstra	Resuelve desde un origen	No pesos negativos	Google Maps
Bellman-Ford	Resuelve desde un origen	trabaja con pesos negativos	Proveedor de servicios de internet
Floyd-Warshall	compara todos los caminos	encuentra todas las rutas mínimas	invertir matrices
Algoritmo A*	elige ruta óptima como dijkstra	resuelve en base a un par de nodos	encontrar Geo localizaciones