

# Java-feb-2016.pdf



Juanma21\_



Estructuras de Datos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática  
Universidad de Málaga

**WUOLAH + BBVA**

## Te regalamos

# 15€



**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

**1**

Abre tu Cuenta  
Online  
sin comisiones  
ni condiciones

**2**

Haz una compra  
igual o superior  
a 15€ con tu  
nueva tarjeta

**3**

BBVA  
te devuelve  
un máximo de  
15€



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

1

Abre tu Cuenta Online sin comisiones ni condiciones

2

Haz una compra igual o superior a 15€ con tu nueva tarjeta

3

BBVA te devuelve un máximo de 15€

File - C:\Users\juanm\OneDrive - Universidad de Málaga\Escritorio\Juanma\aa UMA\2º\Primer Cuatri\ED\Java\Examen\_feb\_20

```

1  /**
2   * Student's name:
3   * Student's group:
4   *
5   * Data Structures. Grado en Informática. UMA.
6   */
7
8  package dataStructures.graph;
9
10 import dataStructures.list.*;
11
12 public class EulerianCycle<V> {
13     private List<V> eCycle;
14
15     @SuppressWarnings("unchecked")
16     public EulerianCycle(Graph<V> g) {
17         Graph<V> graph = (Graph<V>) g.clone();
18         eCycle = eulerianCycle(graph);
19     }
20
21     public boolean isEulerian() {
22         return eCycle != null;
23     }
24
25     public List<V> eulerianCycle() {
26         return eCycle;
27     }
28
29     // J.1
30     private static <V> boolean isEulerian(Graph<V> g
31 ) {
32         if(g.isEmpty() || g.numVertices() == 1){
33             return true;
34         }else if(g.numVertices() == 2){
35             for (V v: g.vertices()) {
36                 if(g.degree(v) % 2 != 0){
37                     return false;
38                 }
39             }
40         }else{
41             for (V v: g.vertices()) {

```

```

41         if(g.degree(v) % 2 != 0){
42             return false;
43         }
44     }
45 }
46     return true;
47 }
48
49 // J.2
50 private static <V> void remove(Graph<V> g, V v, V
    u) {
51     g.deleteEdge(v, u);
52
53     for (V x : g.vertices()) {
54         if(g.degree(x) == 0){
55             g.deleteVertex(x);
56         }
57     }
58 }
59
60 // J.3
61 private static <V> List<V> extractCycle(Graph<V>
    g, V v0) {
62     List<V> ciclo = new LinkedList<>();
63     ciclo.append(v0);
64
65     V v = v0;
66
67     while (g.degree(v) > 0) {
68         V u = g.successors(v).iterator().next();
69         ciclo.append(u);
70         g.deleteEdge(v, u);
71         remove(g, v, u);
72         v = u;
73     }
74
75     return ciclo;
76 }
77
78 // J.4
79 private static <V> void connectCycles(List<V> xs

```

# Te regalamos

**15€**



**1**

Abre tu Cuenta  
Online  
sin comisiones  
ni condiciones

**2**

Haz una compra  
igual o superior  
a 15€ con tu  
nueva tarjeta

**3**

BBVA  
te devuelve  
un máximo de  
15€



```

79 , List<V> ys) {
80     if(xs.isEmpty()){
81         xs = ys;
82         return;
83     }
84
85     int index = -1;
86     V y0 = ys.get(0);
87
88     for (int i = 0; i < xs.size(); i++) {
89         if (xs.get(i).equals(y0)) {
90             index = i;
91             break;
92         }
93     }
94
95     xs.remove(index);
96
97     for (int i = 0; i < ys.size(); i++) {
98         xs.insert(index + i, ys.get(i));
99     }
100
101     //return de xs
102 }
103
104 // J.5
105 private static <V> V vertexInCommon(Graph<V> g,
List<V> cycle) {
106     for (V v: cycle) {
107         if(g.vertices().isElem(v)){
108             return v;
109         }
110     }
111     return null;
112 }
113
114 // J.6
115 private static <V> List<V> eulerianCycle(Graph<V
> g) {
116     if(!isEulerian(g)){
117         return null;

```



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

1

Abre tu Cuenta Online sin comisiones ni condiciones

2

Haz una compra igual o superior a 15€ con tu nueva tarjeta

3

BBVA te devuelve un máximo de 15€

File - C:\Users\juanm\OneDrive - Universidad de Málaga\Escritorio\Juanma\aa UMA\2ºPrimer Cuatri\ED\Java\Examen\_feb\_20

```

118         //throw new GraphException("El grafo no
        es euleriano :(");
119     }
120
121     List<V> cicloEuleriano = new LinkedList<>();
122     List<V> cicloAuxiliar = new LinkedList<>();
123     V v1 = g.vertices().iterator().next();
124     V v2;
125     cicloEuleriano = extractCycle(g, v1);
126
127     while (!g.isEmpty()){
128         v2 = vertexInCommon(g, cicloEuleriano);
129         cicloAuxiliar = extractCycle(g, v2);
130         connectCycles(cicloEuleriano,
        cicloAuxiliar);
131     }
132
133     return cicloEuleriano;
134 }
135 }
136
    
```