SHOPPING PATTERNS

https://leetcode.com/playground/7jQWt4su

// "static void main" must be defined in a public class.

public class Main {

public static void main(String[] args) {

//int score = getMinScore(6,6,new int[]{1,2,2,3,4,5}, new int[]{2,4,5,5,5,6});

int score = getMinScore(5,6,new int[]{1,1,2,2,3,4}, new int[]{2,3,3,4,4,5});

System.out.println("Min score is "+score);

}

public static int getMinScore(int products\_nodes, int products\_edges , int[] products\_from, int[] products\_to)

{

Map<Integer, Set<Integer>> edges = buildMap(products\_from, products\_to);

int minSum=Integer.MAX\_VALUE;

for(int vertex: edges.keySet())

{

if(edges.get(vertex)!=null && edges.get(vertex).size()==2)

{

minSum = Math.min(minSum, findProSum(vertex, edges));

}

}

return minSum;

}

public static int findProSum(int vertex, Map<Integer, Set<Integer>> edges)

{

Set<Integer> trioSet = edges.get(vertex);

trioSet.add(vertex);

int count=0;

for(int i: trioSet)

{

for(int j: edges.get(i))

{

if(!trioSet.contains(j))

{

count++;

}

}

}

return count;

}

public static Map<Integer, Set<Integer>> buildMap(int[] start, int[] end)

{

Map<Integer, Set<Integer>> edges = new HashMap();

for(int i=0;i<start.length;i++)

{

Set<Integer> vertices = edges.getOrDefault(start[i], new HashSet<>());

vertices.add(end[i]);

edges.put(start[i], vertices);

vertices = edges.getOrDefault(end[i], new HashSet<>());

vertices.add(start[i]);

edges.put(end[i], vertices);

}

return edges;

}

}