

PARTE 2: Ejercicios de aplicación
(se puede tener material MANUSCRITO PERSONAL)
Duración: 60 minutos

EJERCICIO 1

ESCENARIO

En el contexto de una pandemia muy agresiva – COVID19 - , se ha desarrollado una aplicación capaz de registrar los contactos entre las personas, a fin de poder, en el caso de que alguna de ellas se haya infectado, realizar tareas de trazabilidad. El sistema almacena los contactos registrados entre dos personas cualquiera, entre las que se asume que ha habido contacto suficiente para que el contagio de la enfermedad pueda existir, en caso de que una de ellas se encuentre cursando la enfermedad.

Persona 1 – Persona 2

Persona 1 – Persona 3

Persona 2 – Persona 3

Persona 3 – Persona 4

A partir de estos datos, una de las funcionalidades que entonces se quiere desarrollar es que, al indicar que una cierta persona se ha verificado como COVID19 positiva, se analicen las vías de trazabilidad hacia todas las demás personas registradas en la red, y se emitan reportes que indiquen qué personas se encuentran en cada nivel de “distancia de contacto” (distancia de contacto 1, 2, etc.). Por ejemplo, en la red con las personas indicadas arriba, si se verifica que la Persona 1 padece la enfermedad, el sistema debe generar diferentes conjuntos, indexados por “distancia de contacto”:

Si maxDistancia es 1:

- Distancia de contacto 1: {Persona2, Persona 3}

Si maxDistancia es 3:

- Distancia de contacto 1: {Persona2, Persona 3}
- Distancia de Contacto 2: {Persona 4}
- Distancia de Contacto 3: { }

TAREA

1. Modela el problema con un TDA adecuado
2. Desarrolla un método de ese TDA que devuelva los conjuntos de personas – “**anillos de contagio**” - que se encuentran a cada “**distancia de contacto**” de la persona indicada como infectada, hasta un **máximo de distancia** de contacto indicado como parámetro.
 - a. En el TDA correspondiente **anillosDeProbablesContagiados** (de tipo Persona *personaCOVID*, de tipo entero **maxDistancia**) devuelve una instancia de la clase “**AnillosContagio**” (en la que, para cada distancia de contacto, se almacena el conjunto de personas correspondiente), para las distancias 1 a **maxDistancia**. Si **maxDistancia** es 0, se deben explorar todos los contactos alcanzables desde la persona indicada.
 - b. La clase **AnillosContagio** exhibe una operación “**agregarContagio**” que recibe como parámetros la distancia de contacto y la Persona correspondiente - **agregarContagio(entero distanciaContacto, Persona laPersona)** -
3. El método debe identificar y registrar en los conjuntos las personas en **forma incremental por distancia de contacto** (en primer lugar todas las de distancia 1, luego las de distancia 2, y así sucesivamente), y tener el mejor orden del tiempo de ejecución posible.
4. Analiza detalladamente el orden del tiempo de ejecución del método desarrollado.

EJERCICIO 2

ESCENARIO

Como método auxiliar para usar en diversos algoritmos relacionados con grafos, es necesario contar con una función que permita decidir, de la forma más eficiente posible, si existe un camino desde cierto vértice origen hasta cierto vértice destino.

TAREA

Desarrollar un método en la clase de `TVertice` que, en una sola recorrida y de la forma más eficiente posible, implemente la siguiente firma

boolean hayCamino (TVertice destino) // devuelve verdadero si existe un camino entre el origen y el destino, y falso en caso contrario.