

Algoritmos y Estructuras de Datos (VJ1215) - Universitat Jaume I

Evaluación continua - 2024/2025

13 de diciembre de 2024

Nombre:

La duración de esta prueba es de 50 minutos. Consta de un solo ejercicio con dos apartados. La prueba es individual. No puedes consultar libros, apuntes ni dispositivos electrónicos. Al finalizar entrega tu solución junto con el enunciado (no es necesario que entregues todas las hojas). Pon tu nombre en cada hoja que entregues.

En cada apartado, debes decidir qué algoritmo utilizar e implementarlo en C++, añadiendo el método que se pide a la siguiente clase utilizada para representar grafos dirigidos en los ejercicios de la asignatura. No puedes añadir ahí otros atributos.

```
class GrafoDirigido {
    struct Arco {
        int vecino;
        float peso;
        Arco * siguiente;
        Arco(int, float, Arco *);
    };
    struct Vertice {
        Arco * primerArcoDeEntrada;
        Arco * primerArcoDeSalida;
        int gradoDeEntrada;
        int gradoDeSalida;
        Vertice();
    };
    vector<Vertice> vertices;
public:
    ...
};
```

EJERCICIO

2,5 PUNTOS

Consideremos un grafo dirigido $G = (V, E)$ que representa los posibles movimientos en un juego. En este juego, quedan n guerreros enfrentándose a un jugador. Cada guerrero está ubicado en un vértice, y puede haber varios guerreros en el mismo vértice. El jugador también está ubicado en un vértice. Un guerrero puede alcanzar al jugador si ambos se encuentran en el mismo vértice o si existe al menos un camino desde el vértice del guerrero hasta el vértice del jugador.

Necesitamos añadir a la clase `GrafoDirigido` un método

```
bool aSalvo(int jugador, int vidas, const vector<int> & guerreros) const
```

que recibe: *jugador*, el vértice en el que se encuentra el jugador; *vidas*, la cantidad de vidas que le quedan al jugador, que sabemos que es un valor positivo; y *guerreros*, un vector de tamaño n que contiene los vértices en los que se encuentran los n guerreros. El método debe devolver *true* si, y solo si, la cantidad de guerreros que pueden alcanzar al jugador (suponiendo que el jugador no cambia de posición) es menor que la cantidad de vidas del jugador. En la siguiente página se proporciona un ejemplo.

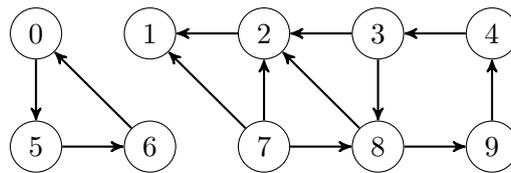
En cada uno de los siguientes dos apartados, tu solución debe ser eficiente para que pueda recibir puntuación. Además, se valorará que no siga recorriendo el grafo cuando ya se pueda determinar el resultado.

- a) [1 punto] Implementa el método `aSalvo` sin utilizar recursión. Si lo necesitas, en este apartado, puedes utilizar las clases `stack`, `queue` y/o `priority_queue` de C++, sin necesidad de implementarlas. Si no recuerdas los nombres de sus operaciones básicas, puedes escribirlos en castellano. Si haces uso de otras funciones, debes implementarlas también sin utilizar recursión.
- b) [1,5 puntos] Implementa el método `aSalvo` utilizando recursión para resolver el problema. Puedes utilizar también bucles. En este apartado, no puedes utilizar las clases `stack`, `queue` ni `priority_queue`, ni implementarlas, ni utilizar, para hacer el papel de esas clases, vectores u otras estructuras de datos.

Indica cuáles son los siguientes costes de tus dos soluciones en función de la cantidad de guerreros n , la cantidad de vértices $|V|$ y la cantidad de arcos $|E|$. No es necesario justificarlos. En el coste espacial, se pide no tener en cuenta lo que ocupa en memoria el propio grafo.

	Apartado a	Apartado b
Coste temporal en el peor caso	$O(\quad)$	$O(\quad)$
Coste espacial en el peor caso sin contar el grafo	$O(\quad)$	$O(\quad)$

Ejemplo



Con este grafo, supongamos que $guerreros = [7, 9, 2, 9, 6]$, lo que indica que hay 5 guerreros distribuidos de la siguiente manera: uno en el vértice 7, uno en el vértice 2, uno en el vértice 6 y dos en el vértice 9. Si el jugador se encontrase en el vértice 8 y dispusiese de 4 vidas, el resultado sería *true*. Esto se debe a que solo 3 guerreros podrían alcanzar al jugador (los situados en los vértices 9 y 7) y esa cantidad es menor que las vidas del jugador. Sin embargo, con las mismas ubicaciones de guerreros y jugador, si el jugador tuviese 3 vidas o menos, el resultado sería *false*.