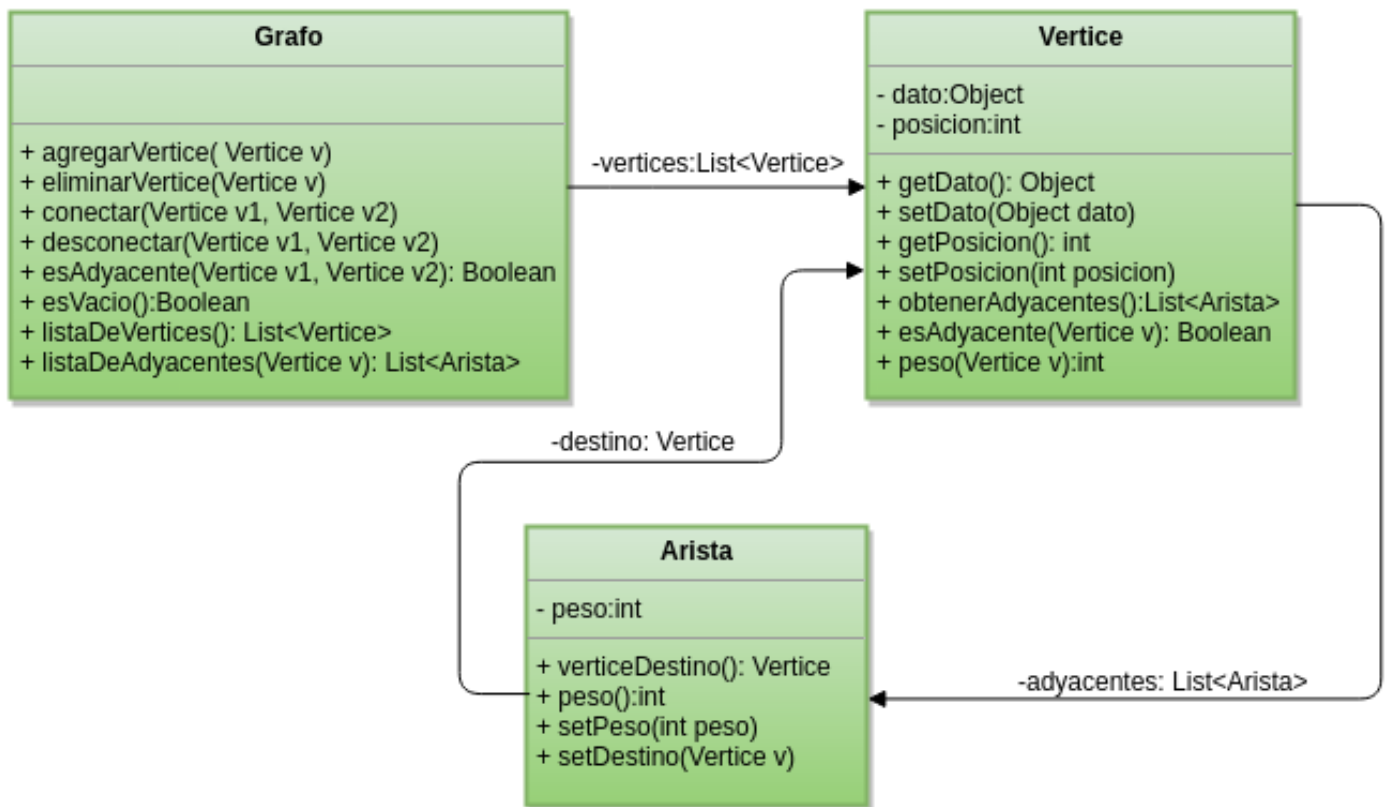


Estructuras de Datos y Algoritmos II UPE 2016

Trabajo Práctico N°3

Fecha parcialito:07/06/2016



1. Implemente en Python una clase llamada Recorridos , cumpliendo la siguiente especificación:

dfs(Grafo grafo): List<Vertice> // Retorna una lista de vértices con el recorrido en profundidad del grafo recibido como parámetro.

bfs(Grafo grafo): List< Vertice> // Retorna una lista de vértices con el recorrido en amplitud del grafo recibido como parámetro.

2.

Mapa
-ciudades: Grafo<String>
+ devolverCamino(String ciudad1, String ciudad2): List<String> + devolverCaminoExceptuando(String ciudad1, String ciudad2, List<String> ciudades): List<String> + caminoMasCorto(String ciudad1, String ciudad2): List<String> + caminoMasLargo(String ciudad1, String ciudad2): List<String>

El método ***devolverCamino (String ciudad1, String ciudad2):List<String>***// Retorna la lista de ciudades que se deben atravesar para ir de ciudad1 a ciudad2 en caso que se pueda llegar, si no retorna la lista vacía.

El método ***devolverCaminoExceptuando (String ciudad1, String ciudad2, List<String> ciudades): List<String>*** // Retorna la lista de ciudades que forman un camino desde ciudad1 a ciudad2, sin pasar por las ciudades que están contenidas en la lista ciudades pasada por parámetro, si no existe camino retorna la lista vacía.

El método ***caminoMasCorto(String ciudad1, String ciudad2):List<String>*** // Retorna la lista de ciudades que forman el camino más corto para llegar de ciudad1 a ciudad2, si no existe camino retorna la lista vacía.

El método ***caminoMasLargo(String ciudad1, String ciudad2):List<String>*** // Retorna la lista de ciudades que forman el camino más largo para llegar de ciudad1 a ciudad2, si no existe camino retorna la lista vacía.

3. Jack desea aprovechar al máximo su estadía en la ciudad de Buenos Aires y antes de emprender su viaje por la Argentina, desea visitar el MACBA (una galería de arte). Como Jack tiene poco tiempo para recorrerla también necesita optimizar el tiempo en ella: el pretende recorrer la mayor cantidad de salas posibles en el tiempo limitado del que dispone.

Considere que la galería está organizada en salas y pasillos que las unen, y que cada sala tiene un tiempo necesario para recorrerla, como así también los pasillos.

Teniendo en cuenta que esta galería está modelada como un grafo, en donde le indican cual es la entrada, implemente un algoritmo para buscar el recorrido que transite la mayor cantidad de salas (sin repetir salas) en 2 horas.

Implemente un método para buscar el mejor recorrido.

4. Sea una red de empresas, en donde una empresa brinda servicios a muchas otras empresas sin ningún tipo de restricciones. Esta relación no es simétrica, es decir, la empresa A puede brindar servicios a la empresa B, sin embargo, B no necesariamente debe brindar servicios a la empresa A.

Usted es contratado por una nueva empresa que quiere ingresar a la Red. Por lo cual, le interesará determinar las 5 empresas de la red que llegan a la mayor cantidad de empresas. Una empresa A llega a otra empresa B, si le brinda servicios directa o indirectamente a través de otras empresas.

5. Resuelva las siguientes funciones de recurrencias calcular su tiempo de ejecución y el orden justificando por Big-Oh.

<p>a)</p> $T(n) = \begin{cases} 1 & \text{si } n = 1 \\ 8T(n/2) + n^3 & \text{si } n \geq 2 \end{cases}$	<p>b)</p> $T(n) = \begin{cases} c & \text{si } n = 0 \\ d & \text{si } n = 1 \\ 2T(n/2) + n^2 & \text{si } n \geq 2 \end{cases}$
<p>c)</p> $T(n) = \begin{cases} 2 & \text{si } n = 1 \\ T(n-1) + n & \text{si } n \geq 2 \end{cases}$	<p>d)</p> $T(n) = \begin{cases} 2 & \text{si } n = 1 \\ T(n-1) + n/2 & \text{si } n \geq 2 \end{cases}$