

ESTRUCTURAS DE DATOS

TRABAJO PRÁCTICO N ° 9
Grafos

Departamento de Ciencias e Ingeniería de la Computación - U.N.S.
Primer cuatrimestre de 2019

Bibliografía:

- [GT] Michael Goodrich & Roberto Tamassia. *Data Structures and Algorithms in Java. Fourth Edition* . John Wiley and Sons. 2006.
- [A] Alfred V. Aho, J. D. Ullman, J. E. Hopcroft. *Estructuras de datos y algoritmos*. Addison-Wesley 1983.
- [W] Mark A. Weiss. *Estructuras de datos* . Addison-Wesley Iberoamericana España, S.A. 2000.
-

Ejercicio 1: Grafos no dirigidos representados con matriz de adyacencias

- Defina el TDA grafo con vértices y arcos pesados de acuerdo a [GT] y documente apropiadamente las operaciones.
- Implemente en Java el TDA grafo utilizando una matriz de adyacencias. Analice variantes para la representación.
- ¿Cómo modificaría la interfaz definida en (a), y la clase implementada en (b), para implementar un multigrafo (es decir, se permite tener más de un arco entre dos vértices)?
- Determine para cada una de las operaciones implementadas en (b) el orden del tiempo de ejecución.
- Determine para la estructura de datos propuesta en (b) el espacio de memoria utilizado para representar un grafo con n vértices y m arcos.

Ejercicio 2: Grafos no dirigidos representados con lista de adyacencias

- Implemente en Java el TDA grafo utilizando una lista de adyacencias. Analice variantes para la representación.
- ¿Cómo modificaría la interfaz definida en (a), y la clase implementada en (b), para que permitan tener más de un arco entre dos vértices?
- Determine para cada una de las operaciones implementadas en (b) el orden del tiempo de ejecución.
- Determine para la estructura de datos propuesta en (b) el espacio de memoria utilizado para representar un grafo con n vértices y m arcos.

Ejercicio 3:

- Implemente en Java un recorrido en profundidad (Depth-first search) sobre un grafo. Analice dos variantes: (i) la operación de recorrido es parte de la clase grafo, y (ii) la operación de recorrido no pertenece a la clase grafo.
- Indique el orden del tiempo de ejecución del recorrido implementado en (a) ¿Influye la representación del grafo en el tiempo de ejecución de los recorridos?

Ejercicio 4:

- a) Implemente en Java un recorrido en anchura (Breadth-first search) sobre un grafo. Analice dos variantes: (i) la operación de recorrido es parte de la clase grafo, y (ii) la operación de recorrido no pertenece a la clase grafo.
- b) Indique el orden del tiempo de ejecución del recorrido implementado en (a) ¿Influye la representación del grafo en el tiempo de ejecución de los recorridos?

Ejercicio 5:

- a) Implemente en Java un algoritmo que dado un grafo con arcos pesados y dos vértices A y B, encuentre el camino más económico de A a B. Determine el orden del tiempo de ejecución de su solución considerando la complejidad temporal de la implementación del grafo y de cualquier otra clase que necesitara.
 - b) Implemente en Java un algoritmo que dado un grafo y dos vértices A y B, encuentre el camino más corto de A a B. Determine el orden del tiempo de ejecución de su solución considerando la complejidad temporal de la implementación del grafo y de cualquier otra clase que necesitara.
 - c) Modifique la solución dada en el inciso (b) para hallar todos los caminos de A a B.
 - d) Modifique la solución dada en el inciso (b) para hallar al menos un camino de A a B.
- Compare el orden del tiempo de ejecución entre las soluciones (c) y (d). Para ello suponga que la estructura de datos subyacente para el grafo es (i) una matriz de adyacencias y (ii) una lista de adyacencias.

Ejercicio 6:

- a) Discuta qué modificaciones es necesario hacer en el TDA Grafo presentado en el ejercicio (1) para implementar grafos dirigidos (o digrafos).
- b) Escriba un método que, dado un rótulo R y un digrafo G, encuentre el primer vértice cuyo rótulo es R.
- c) Escriba un método que recorriendo el digrafo G en profundidad, busque el camino más largo (sin ciclos) para llegar del vértice V1 al vértice V2. Indique el orden del tiempo de ejecución del procedimiento implementado.
- d) Escriba un método que reciba un digrafo G y el rótulo R de un vértice, y que elimine de G todos los vértices que encuentre con rótulo R.
- e) Escriba un método que reciba un digrafo G y un vértice A, y establezca si G es una lista cabeza A.
- f) Escriba un método que reciba un digrafo G y un vértice A, y establezca si G es un árbol con raíz A.
- g) Escriba un método que, dado un digrafo, busque cuál es el vértice con mayor grado de incidencia (es decir, al cual apuntan la mayor cantidad de arcos), y luego lo elimine.

Ejercicio 7:

- a) Defina una clase DigrafoCiudadesUnidasConRutas donde la información de cada vértice representa una ciudad y las aristas representan rutas entre las ciudades. Considere que las aristas se hallan pesadas con la distancia entre cada par de ciudades.
- b) Escriba un método que reciba como parámetros los nombres de tres poblaciones P1, P2 y P3 y devuelva en C (una lista de Poblaciones) el camino, si es que existe, que permita ir de P1 a P2 pasando por P3.
- c) Escriba un método para la clase DigrafoCiudadesUnidasConRutas, que, con total acceso a la estructura de datos, reciba por parámetros los nombres de dos poblaciones P1 y P2 y que, recorriendo el grafo en profundidad, cuente la cantidad de caminos (sin ciclos) para llegar de P1 a P2, y que determine la distancia en kilómetros más corta de P1 a P2.
- d) Indique el orden del tiempo de ejecución de los procedimientos implementados. Justifique.

Ejercicio 8:

- a) Escriba en Java un método para determinar si un grafo es conexo.
- b) Escriba en Java un método que determine las componentes conexas de un grafo.
- c) Realice un método que elimine un vértice y determine si varió el número de componentes conexas del grafo.

Ejercicio 9:

- a) Implemente el algoritmo de Warshall.
- b) Utilizando el algoritmo de Warshall, encuentre la clausura reflexo-transitiva de la relación R de enteros en enteros tal que:
 $R = \{ (1,1); (1,2); (1,7); (2,3); (3,4); (4,5); (4,7); (5,6); (8,7); (4,9) \}$

Ejercicio 10:

- a) Implemente el algoritmo de Floyd para digrafos pesados donde el peso de las aristas son números reales.
- b) Utilice la implementación realizada en (a) para resolver nuevamente los incisos (5.a) y (5.b).
- c) Considere cómo resolver el mismo problema pero implementando previamente el algoritmo de Dijkstra.
- d) Analice el orden del tiempo de ejecución de las soluciones (b) y (c) y compárelas con las dadas en (5.a) y (5.b).