

Taller en Sala Nro. 11 Programación Dinámica para el Agente Viajero



En la vida real se necesita resolver el problema del agente viajero para la construcción de circuitos electrónicos, entrega de domicilios, ruteo de aviones y recolección de las monedas de los teléfonos públicos.



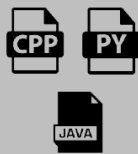
Trabajo en
Parejas



Hoy, plazo
máximo de
entrega



Docente entrega
código suelto en
GitHub



Sí .cpp, .py
o .java



No .zip, .txt,
html o .doc



Alumnos
entregan
código suelto
por GitHub

Ejercicio a resolver

1. Dado un grafo completo halle el costo mínimo del recorrido que pasa por todos los vértices exactamente una vez y vuelve al inicial **utilizando programación dinámica** (el algoritmo de Held-Karp).

```
public static int heldKarp(Digraph g) {  
    // complete...  
}
```

Ayudas para resolver el Ejercicio



Pista 1: Es el mismo problema que hemos trabajado antes (tanto con *Backtracking* como con la técnica *Greedy* del vecino más cercano), solo que esta vez lo resolveremos de manera exacta y mediante la forma más eficiente conocida.

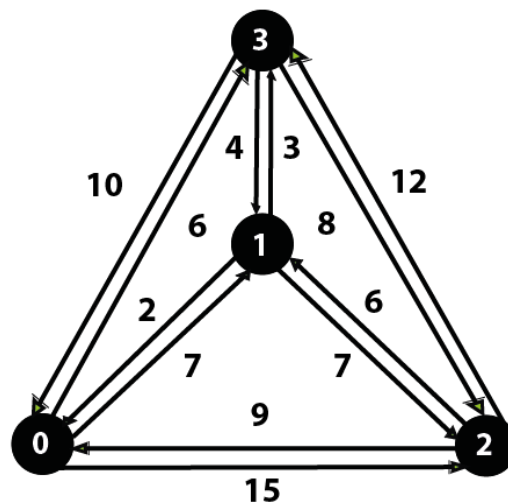


Pista 2: Para que la implementación sea lo más eficiente posible debe representar los conjuntos utilizando máscaras de bits.

En caso de que tenga problemas con las máscaras de bits puede utilizar la clase *BitmaskSet* que hemos creado para usted, la cual pretende abstraer las máscaras de bits en forma de *Set* de Java (aunque con muchas limitaciones, pero ideal para la implementación de este algoritmo). Si deciden usarla, recuerden leer la documentación.



Como un ejemplo, en el siguiente grafo el costo total del recorrido de costo mínimo que pasa por los cuatro vértices (iniciando y volviendo a 0) de 22:



Iniciando en 0, esta es la tabla de ejecución del algoritmo:

	Costo	Padre
[1, {}]	7	0
[2, {}]	15	0
[3, {}]	6	0
[2, {1}]	14	1
[3, {1}]	10	1
[1, {2}]	21	2
[3, {2}]	27	2
[1, {3}]	10	3
[2, {3}]	14	3
[3, {1, 2}]	$\text{Min}(3 + [1, \{2\}], 12 + [2, \{1\}]) = \text{Min}(24, 26) = 24$	1
[1, {2, 3}]	$\text{Min}(6 + [2, \{3\}], 4 + [3, \{2\}]) = \text{Min}(20, 31) = 20$	2
[2, {1, 3}]	$\text{Min}(7 + [1, \{3\}], 8 + [3, \{1\}]) = \text{Min}(17, 18) = 17$	1
[0, {1, 2, 3}]	$\text{Min}(2 + [1, \{2, 3\}], 9 + [2, \{1, 3\}], 10 + [3, \{1, 2\}]) = 22$	1

Reconstruyendo tenemos $0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$.



Pista 3: Si tiene problemas para la implementación del algoritmo le recomendamos el siguiente video: <https://www.youtube.com/watch?v=-JjA4BLQyqE>. Recuerden activar los subtítulos (en inglés) en caso de que los necesite

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 **Ext.** 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agende una cita con él a través de <http://bit.ly/2gzVg10> , en la pestaña *Semana*. *Si no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.*