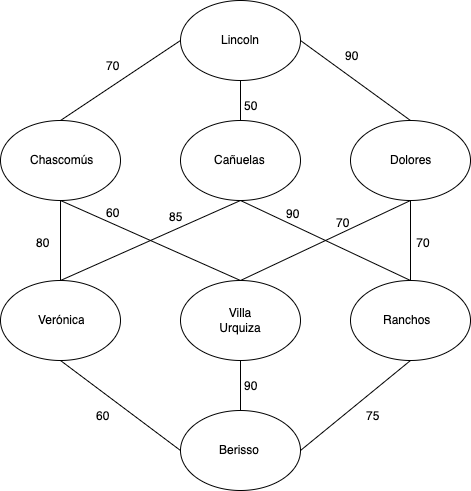
**Algoritmos y Estructuras de datos - Curso 2022 - Parcial módulo 2 - Grafos**

**Sábado 2 de Julio de 2022 - Tema 1**

**Ejercicio 1 -- 5 Puntos**

Se cuenta con información del precio de los peajes para transitar por las rutas que unen las distintas ciudades. Se quiere obtener **un camino cualquiera** que comience en una ciudad origen y permita llegar a otra ciudad destino sin sumar más de X pesos en el total a pagar en peajes. En caso de no existir camino deberá retornar una lista vacía.



Para el grafo del ejemplo, considerando que el origen es Lincoln, el destino Berisso y el monto máximo es 200, un camino posible es:

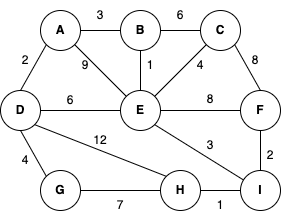
**Lincoln > Cañuelas > Verónica > Berisso**, ya que el costo de los peajes suman 195 (50+85+60).

Implemente en la clase BuscadorDeCamino el método:

**ListaGenerica<String> caminoConPresupuesto(Grafo<String> ciudades, String origen, String destino, int montoMaximo)**

**Ejercicio 2 -- 3 Puntos**

Se desea ejecutar el algoritmo de Dijkstra sobre el siguiente di-grafo pesado, a partir del vértice “A”.

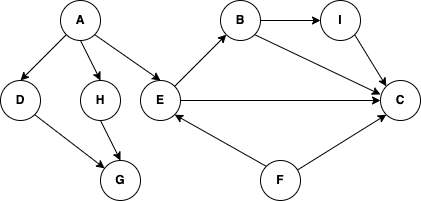
****

Muestre todos los pasos intermedios, indicando el orden en que se van procesando los vértices.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orden en el que  toma el vértice | Vértice V | Distancia (A,V) | Previo | Visitado |
| 1º | A | ~~∞~~ 0 | - | ~~0~~ 1 |
|  | B | ∞ |  | 0 |
|  | C | ∞ |  | 0 |
|  | D | ∞ |  | 0 |
|  | E | ∞ |  | 0 |
|  | F | ∞ |  | 0 |
|  | G | ∞ |  | 0 |
|  | H | ∞ |  | 0 |
|  | I | ∞ |  | 0 |

**Ejercicio 3 -- 2 Puntos**

Indicar cuál es la ordenación topológica para el siguiente grafo dirigido acíclico, utilizando la estrategia que trabaja con un recorrido DFS comenzando por el vértice A (tanto los vértices y los adyacentes se procesan en forma ascendente).



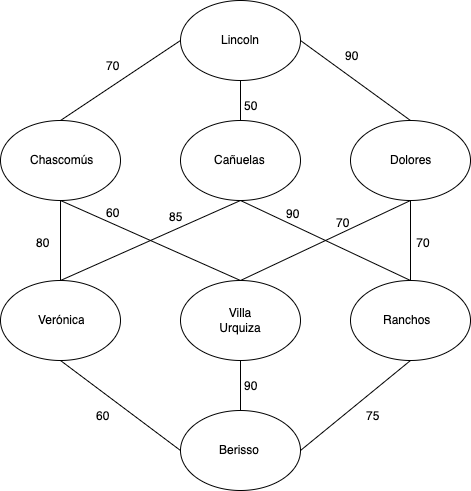
|  |
| --- |
|  |

**Algoritmos y Estructuras de datos - Curso 2022 - Parcial módulo 2 - Grafos**

**Sábado 2 de Julio de 2022 - Tema 3**

**Ejercicio 1 -- 5 Puntos**

Se cuenta con información del precio de los peajes para transitar por las rutas que unen las distintas ciudades. Se quiere obtener **un camino cualquiera** que comience en una ciudad origen y permita llegar a otra ciudad destino sumando más de X pesos en el total a pagar en peajes. En caso de no existir camino deberá retornar una lista vacía.



Para el grafo de ejemplo, considerando que el origen es Lincoln, el destino es Berisso y el monto mínimo es 200 un camino posible es:

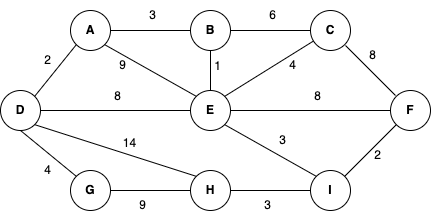
**Lincoln > Dolores > Ranchos > Berisso**, ya que el costo de los peajes suma 235 (90+70+75).

Implemente en la clase BuscadorDeCamino el método:

**ListaGenerica<String> caminoConPresupuesto(Grafo<String> ciudades, String origen, String destino, int montoMinimo)**

**Ejercicio 2 -- 3 Puntos**

Se desea ejecutar el algoritmo de **Dijkstra** sobre el siguiente di-grafo pesado, a partir del vértice **“D”**.

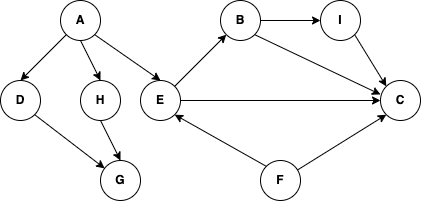


Muestre todos los pasos intermedios, indicando el orden en que se van procesando los vértices.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orden en el que  toma el vértice | Vértice V | Distancia (A,V) | Previo | Visitado |
|  | A | ∞ |  | 0 |
|  | B | ∞ |  | 0 |
|  | C | ∞ |  | 0 |
| 1º | D | ~~∞~~ 0 | - | ~~0~~ 1 |
|  | E | ∞ |  | 0 |
|  | F | ∞ |  | 0 |
|  | G | ∞ |  | 0 |
|  | H | ∞ |  | 0 |
|  | I | ∞ |  | 0 |

**Ejercicio 3 -- 2 Puntos**

Obtener la **ordenación topológica** para el siguiente grafo dirigido acíclico, utilizando la estrategia que trabaja con los grados de entrada cero de los vértices almacenados en una **cola**. Muestre la ejecución del algoritmo indicando **cada paso** cómo se van modificando los grados de entrada de los vértices. *Nota*: tantos los vértices como los adyacentes se procesan en forma ascendente.



|  |
| --- |
|  |