

## Carreras: Licenciatura e Ingeniería en Sistemas

### Parcial 2 de: Estructuras de datos y Algoritmos 2

Código de materia:

Fecha: 5-12-19

Id Examen:

Acta:

Hoja 1 de 1

#### Ejercicio 1 (25 puntos)

Una persona que vive en una localidad montañosa desea trasladarse de un pueblo a otro de la manera menos costosa posible. A igual costo, desea el camino de menor número de pasos. En los últimos días se produjeron nevadas muy severas que dejaron aislados algunos de los pueblos de la zona. Se conoce el costo de ir, directamente, de un pueblo a otro, cuando esto es posible, (en caso de que no haya camino directo de un pueblo a otro se asumirá costo INFINITO) y se posee una lista de los pueblos aislados. Dado que la localidad presenta desniveles, algunos costos serán negativos (ya que aumentará la velocidad sin consumir combustible) y otros positivos.

Considere los pueblos numerados con naturales en el rango 1..N.

- a) Se pide resolver el problema, usando la técnica de diseño de algoritmos con retroceso (backtracking) e implementar la función

**Lista**<Lista<int>\*>\*

**caminosMasCortos**(int ciudadOrigen, int ciudadDestino, int numeroDeCiudades, Grafo<int,int>\* g)

Que retorne una lista de el(los) camino(s) más corto(s) de la ciudad origen a la ciudad destino.

El desarrollo de esta función debe incluir un código de primer nivel en el cual se encuentren claramente expresados los aspectos centrales de la técnica aplicada como, por ejemplo, estudio de la factibilidad de la solución, poda (si corresponde), construcción de la solución, consideración de las alternativas disponibles, retroceso.

Serán muy importantes en la evaluación aspectos como legibilidad, uso de nombres auto explicativos para variables, funciones, etc.

Se podrán usar funciones auxiliares que se deberán implementar.

**Para este ejercicio puede usar TADS que deberá especificar, pero no implementar.**

#### Ejercicio 2 (20 puntos)

- a) Se pide desarrollar, utilizando la técnica de diseño de algoritmos dividir para conquistar, una función, recibiendo un array de enteros, retorne la suma de los múltiplos de **k** que encuentran entre las posiciones del array **desde** y **hasta** inclusive.

**int sumaMultiplos**(int v[], int desde, int hasta, int k)

5	3	8	9	7	27	14	4	12	16
0	1	2	3	4	5	6	7	8	9

#### Ejemplo

desde	hasta	k	retorno
0	2	5	0
0	2	3	3
2	8	4	3
0	9	3	4
3	3	3	1
3	3	9	1
3	3	7	0

En el algoritmo desarrollado deberá poder apreciarse claramente cuál es el tamaño de problema fácil y cómo se resuelve.

Cómo se divide el problema difícil, cómo se resuelven los subproblemas y cómo se combinan las soluciones a los subproblemas para producir la solución del problema original.

- b) ¿De qué orden es el algoritmo desarrollado en a)?