Parcial de Programación 3 - TUDAI 07/06/2019

Ejercicio 1

Dado un grafo rotulado no dirigido aciclico, y los vértices v, w, z dados como parámetro, y los siguientes problemas:

a) Obtener el **camino más largo** entre el vértice v y el vértice w que no pase por el vértice z.
b) Obtener el **camino más corto** entre el vértice v y el vértice w que no pase por el vértice z.

Diga para cada uno de los problemas:

- Con qué técnica algorítmica lo resolvería para obtener de manera eficiente la solución óptima. Y describa las características principales de dicha/s técnica/s.
- Escriba en **pseudo-código** el algoritmo para cada problema.

Ejercicio 2

Problema de las Permutaciones.

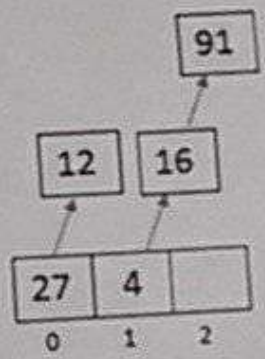
Se tienen N elementos distintos (por ejemplo, una lista con los números 1, 2, y 3) y se quiere obtener todas las formas distintas de colocar esos elementos, es decir, hay que conseguir todas las permutaciones de los N elementos.

a) Muestre gráficamente cómo se construye el árbol de búsqueda.
b) Escriba en **JAVA** un algoritmo que use Backtracking para resolver el problema.

Por ejemplo, para los números {1, 2, 3} se deben obtener las siguientes permutaciones {1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}.

Ejercicio 3

Se tiene una estructura de Hashing separado con crecimiento lineal donde ya se ingresaron los siguientes elementos: 27, 4, 16, 12, 91.



El ρ de diseño es de 1,7
M=3, rp=1, rs=1 y la frontera f se encuentra en 0.
h(x) = x mod M

a) Ingresar el 8 y mostrar la estructura resultante. ¿Cuál es el ρ de carga resultante?. Luego ingresar el elemento 6 y mostrar la estructura que se obtiene. ¿Cuál es el ρ de carga resultante?

Ejercicio 4

Ordenar de MAYOR a MENOR el siguiente arreglo utilizando Merge-Sort: **6-5-3-1-8-7-2-4-10**
(El resultado será el arreglo 10-8-7-6-5-4-3-2-1)

a) Mostrar gráficamente cómo será la ejecución (seguimiento) del algoritmo.
b) Escriba un pseudocódigo para realizar el ordenamiento merge-sort que se pide.