UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA
INFORMÁTICA

Algoritmos Avanzados

Laboratorio N°2: Cálculo de complejidad

Alumnos: Israel Arias Panez.
Christian Mendez Acosta.

Sección: A-1.

Santiago -
Chile 1-2021

Pseudocódigo el programa de goloso:

1. Se Solicita el nombre del archivo por teclado.

2. Se verifica si existe el dataset según el nombre entregado, en caso de no existir vuelve al paso 1.

3. Se almacena la capacidad máxima del container por el nombre del archivo.

4. Se almacena la cantidad de paquetes por el nombre del archivo.

5. Se abre el archivo dataset (archivo .csv).

6. Lectura, validación y creación de paquetes, creación del arreglo inicial que contiene todos los paquetes leídos y validados.

7. Se verifica en caso de que no se haya guardado ningún paquete válido.

8. Realizar un ordenamiento por **Quicksort** de la lista de paquetes de mayor a menor según su ponderación, usando la función nativa qsort(); disponible en la librería estándar de C <stdio.h>.

9. Debido a que la lista de paquetes se encuentra ordenada de mayor a menor según su ponderación, se calcula mediante el método goloso, el índice hasta el cual es posible incorporar paquetes a la solución final sin exceder el volumen máximo que soporta el container.

10. Se escribe un archivo de salida .txt que posee la información de la solución encontrada, la cual fue determinada en el paso 9 según el índice de tope de la lista de paquetes calculado, el cual no excede el volumen máximo del container. Se escribirán los paquetes de la solución encontrada, incluyendo el beneficio y el volumen que ocupa.

Cálculo de complejidad:

Los pasos 1, 2, 3, 4 y 5 son constantes O(1), por lo que no serán calculados a continuación.

6.

Se crea un arreglo de paquetes iniciales vacío
 	
Por cada línea en el archivo:
  Se lee volumen del Paquete de la línea (segundo dígito)
  Si volumen Paquete < Capacidad Container:
  Calcular ponderación -> beneficio/volumen
  Generar paquete según estructura
  Guardar paquete en arreglo inicial
Cerrar archivo
	O(n)

7.

Si largo arreglo inicial == 0: Mostrar por pantalla “No hay solución válida” Fin de Programa	O(1)

8.

Quicksort es un algoritmo de ordenamiento muy usado que tiene una complejidad de O(n log n)	O(n log n)

9.

Se inicializa Volumen Acumulado igual a cero. Para cada elemento del arreglo de paquetes iniciales: Volumen Acumulado = Volumen Acumulado + Volumen elemento Si Volumen Acumulado > Capacidad Container: Retornar índice del ciclo -1 Retornar largoArreglo - 1	O(n)

10.

Abrir archivo con nombre salida.txt Se inicializan las variables beneficioTotal y volumenTotal ambas en cero Para cada elemento de la lista de paquetes inicial hasta indice de tope: Leer numero de id del elemento actual Escribir “Paquete” + id en archivo beneficioTotal = beneficioTotal + beneficio elemento volumenTotal = volumenTotal + volumen elemento Escribir “Beneficio: ” + beneficioTotal en archivo Escribir “Volumen: “ + volumenTotal en archivo Cerrar archivo Fin de Programa	O(n)

Complejidad total

O(n) + O(1) + O(n log n) + O(n) + O(n)

Por cota superior: $n\,log\,n \geq n \geq 1$ entonces la complejidad del algoritmo es de
O(n log n).