

ALGORITMOS Y ESTRUCTURAS DE DATOS III
Trabajo Práctico Nº1

Primera entrega: 18-ABR-2008, hasta las 19:30 horas.

Segunda entrega: 02-MAY-2008, hasta las 19:30 horas.

Ver información general sobre los Trabajos Prácticos en la página de la materia en Internet.

- Desarrollar e implementar algoritmos para los problemas enumerados.
- Utilizando el pseudocódigo, calcular el orden de la complejidad de cada algoritmo, y decir si es constante, logarítmica, lineal, polinomial, exponencial o mayor **en función del tamaño de la entrada**.
- Aplicar los algoritmos a un conjunto de instancias de entrada, midiendo para cada instancia el tiempo de ejecución o la cantidad de operaciones. Elegir las instancias de manera tal que se pueda apreciar el comportamiento de los algoritmos e incluirlas en el soporte digital a entregar.
- Graficar para cada algoritmo el tiempo de ejecución o la cantidad de operaciones en función del *tamaño de la entrada*. **Comparar con la complejidad teórica calculada.**
- Leer los datos de entrada desde un archivo con el nombre `Tp1EjX.in`, donde `X` es el número del problema. Escribir los resultados en un archivo con el nombre `Tp1EjX.out`.
- Respetar los formatos de archivos que se indican en cada caso.
- Ignorar los costos de lectura y escritura de los archivos tanto al medir como al calcular complejidad.

1. Enunciado

Dado un número natural n mayor que 1, encontrar el número primo p que aparece con mayor potencia en la factorización de n . En caso de haber más de un número primo con la mayor potencia, encontrar el mayor de ellos.

Entrada `Tp1Ej1.in`

Cada instancia de entrada es definida en una línea conteniendo el número n . Cada valor puede estar precedido o seguido por una cantidad arbitraria de espacios en blanco. El archivo conteniendo varias instancias termina con una línea conteniendo un 0, la cual no debe ser considerada como una instancia.

Salida `Tp1Ej1.out`

Para cada instancia de la entrada, se debe escribir una línea en la salida conteniendo el número n , el primo p y la potencia en la factorización, separados por un sólo espacio en blanco.

Ejemplo

<code>Tp1Ej1.in</code>	<code>Tp1Ej1.out</code>
7	7 7 1
25	25 5 2
225	225 5 2
135	135 3 3
0	

2. Enunciado

La empresa Musimundo cuenta con una serie de sucursales repartidas por el país. Recientemente ha decidido cerrar su sucursal de Iruya y llevarse toda la mercadería al depósito central. Debido al difícil acceso ha dispuesto sólo un camión para llevar toda la mercadería. Sin embargo, es posible que no todo el material pueda ser llevado en un sólo viaje del camión debido a las restricciones de carga, por lo que la parte que no entre en el camión será vendida a valores despreciables el día del cierre.

Dada la capacidad de carga máxima P del camión y una lista de los productos del local conteniendo el valor v_i y peso p_i de cada uno, encontrar la lista de productos de mayor valor que sea posible llevar en el camión sin que el peso total de la lista supere la carga máxima. El valor de una lista de productos se calcula como la suma de los valores de los productos involucrados. En caso de haber varias listas con el mismo valor máximo, encuentre cualquiera de ellas.

Realice un algoritmo para resolver el problema usando la técnica de *backtraking*.

Entrada Tp1Ej2.in

Cada instancia de entrada es definida en $n + 2$ líneas, siendo n la cantidad de productos. La primera línea de la instancia contiene solamente la palabra “Caso”. La segunda línea de la instancia cuenta con dos enteros positivos, n y P . En cada una de las siguientes n líneas hay dos enteros positivos v_i y p_i que describen el i -ésimo producto de la lista. Cada valor puede estar precedido o seguido por una cantidad arbitraria de espacios en blanco y dos valores en una misma línea están separados por al menos un espacio. El archivo conteniendo varias instancias termina con una línea conteniendo solamente “Fin”, la cual no debe ser considerada como una instancia.

Salida Tp1Ej2.out

Para cada instancia de la entrada, se debe escribir una línea en la salida conteniendo el valor total de la lista, la cantidad m de elementos de la lista elegida y la lista de los m índices elegidos, en orden creciente, siendo 1 el índice del primer producto dado en la entrada. Todo par de valores consecutivos en una línea debe estar separado por exactamente un espacio en blanco y no debe haber espacios al comienzo o al final de la línea.

Ejemplo

Tp1Ej2.in	Tp1Ej2.out
Caso	10 1 1
1 50	11 2 2 3
10 50	0 0
Caso	30 3 1 2 3
3 100	
10 51	
6 50	
5 50	
Caso	
2 100	
10 200	
20 150	
Caso	
3 100	
10 20	
10 20	
10 20	
Fin	

3. Enunciado

Dado un arreglo de n elementos en el que hay un elemento que aparece más de la mitad de las veces, encontrar la moda, es decir, el valor que aparece más veces.

Entrada Tp1Ej3.in

Cada instancia de entrada es definida en dos líneas. La primera línea contiene el entero positivo n . La siguiente línea contiene n enteros positivos describiendo el arreglo. Cada valor puede estar precedido o seguido por una cantidad arbitraria de espacios en blanco y dos valores en una misma línea están separados por al menos un espacio. El archivo conteniendo varias instancias termina con una línea con solamente un 0, la cual no debe ser considerada como una instancia.

Salida Tp1Ej3.out

Para cada instancia de la entrada, se debe escribir una única línea en la salida conteniendo la moda del arreglo.

Ejemplo

Tp1Ej3.in	Tp1Ej3.out
3	1
44 1 1	5
1	9
5	
7	
9 1 9 1 9 1 9	
0	