

Packets
Johanna Beltran y Diego Triviño
2012

Índice

1. Introducción	2
2. Definición del problema	2
2.1. Entrada	3
2.2. Salida	3
3. Modelamiento matemático	3
4. Planteamiento de la solución	4
4.1. Pseudocódigo propuesto	4
5. Conclusiones	4

1. Introducción

'Packets' es un problema de programación el cual encontramos en el juez virtual UVA con el número **311**. Este documento busca mostrar una de las tantas soluciones desde el enfoque matemático teniendo en cuenta que el objetivo es realizar el código con la solución del problema en cualquier lenguaje de programación con la ayuda de este documento.

Este problema puede ser resuelto utilizando la metodología de algoritmos voraces.

Los algoritmos voraces tienden a ser bastante eficientes y pueden implementarse de forma relativamente sencilla. Su eficiencia se deriva de la forma en que trata los datos, llegando a alcanzar muchas veces una complejidad de orden lineal.

Se deben definir los siguientes elementos según el problema que abordan:

1. **El conjunto C :** de candidatos es decir las entradas del problema.
2. **Función solución S :** Comprueba, en cada paso, si el subconjunto actual de candidatos elegidos forma una solución (no importa si es óptima o no lo es).
3. **Función de selección:** Informa de cuál es el elemento más prometedor para completar la solución. Éste no puede haber sido escogido con anterioridad. Cada elemento es considerado una sola vez.
4. **Función de factibilidad:** determina si un conjunto es completable, es decir, si añadiendo a este conjunto nuevos candidatos es posible alcanzar una solución al problema, suponiendo que esta exista.
5. **Función objetivo:** Es aquella que queremos maximizar o minimizar, el núcleo del problema.

2. Definición del problema

Una fábrica produce productos envasados en paquetes cuadrados de la misma altura h y de los tamaños 1×1 , 2×2 , 3×3 , 4×4 , 5×5 , 6×6 . Estos productos se suministran siempre a los clientes en las parcelas cuadradas de la misma altura h y del tamaño 6×6 ; por lo tanto, se debe encontrar el mínimo número de parcelas necesarias para entregar los productos dados de acuerdo a una orden.

Se deben tener en cuenta las siguientes restricciones para la solución del problema:

1. Para cada caso de prueba se deben ingresar necesariamente 6 números enteros.

2.1. Entrada

El archivo de entrada consta de varias líneas que especifican las órdenes. Cada línea especifica una orden. Las órdenes son descritos por seis números enteros separados por un espacio que representa sucesivamente el número de paquetes de tamaño individual, del tamaño más pequeño 1×1 para el tamaño más grande 6×6 . El final del archivo de entrada se indica por una línea que contiene seis ceros.

EJEMPLO

```
0 0 4 0 0 1
7 5 1 0 0 0
0 0 0 0 0 0
```

2.2. Salida

El archivo de salida contiene una línea para cada línea en el archivo de entrada. Esta línea contiene el número mínimo de las parcelas en las que puede ser el fin de la línea correspondiente del archivo de entrada para llevar. No hay una línea en el archivo de salida correspondiente a la última línea “” nula del archivo de entrada.

EJEMPLO ANTERIOR

```
2
1
```

3. Modelamiento matemático

M es la parcela que vamos a llenar con los paquetes que tiene un espacio de 6×6 unidades y cada vez que no quepan más se creara una nueva y se irán contando el número de parcelas que utilizamos.

1. **Conjunto de candidatos C :** Conjunto de paquetes ingresados.
2. **Función de selección (*SelezionePaquete*):** El paquete más grande del conjunto de candidatos que se puede pepetrar en la parcela M .
3. **Función solución (*objetivo*):** El conjunto de candidatos es vacío.

4. Planteamiento de la solución

Para cada caso de prueba se definen los elementos de los algoritmos voraces mencionados anteriormente. Para empezar, con la función de selección se elige un paquete x y se elimina del conjunto C ; si no existe ningún elemento en el conjunto de candidatos que cumpla con los requisitos de selección se aumenta el número de parcelas utilizadas y se crea una nueva. Por último, se verifica la función solución devolviendo así la respuesta correcta para cada caso de prueba.

4.1. Pseudocódigo propuesto

Funcion Voraz (lista C , parcela M)

Define entero $contadorParcelas = 0$

mientras $\neg objetivo(C)$

$x := \text{SeleccionePaquete}(C, M)$

si ($x == null$)

entonces $contadorParcelas := contadorParcelas + 1$

$M := \text{nuevaParcela}()$

sino

entonces $C := C / \{X\}$

$\text{agreguePaquete}(M, X)()$

devuelve $contadorParcela$

5. Conclusiones

1. Por las características de este problema se recomienda utilizar el enfoque de algoritmos voraces puesto que suelen ser bastante rápidos, encontrándose dentro de órdenes de complejidad polinomiales dependiendo del número de candidatos, de las funciones básicas a utilizar y del número de elementos de la solución.
2. En este caso específico la función factibilidad se decidió eliminar, puesto que cualquier elemento de la función selección es factible.