

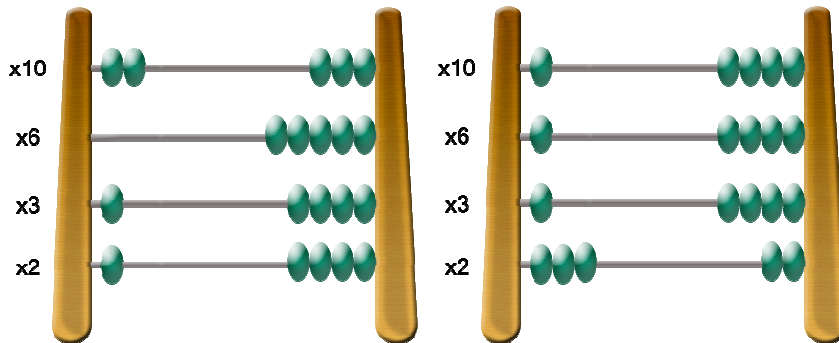
Universidad de la Habana, Ciencia de la Computación

EXAMEN Extraordinario de PROGRAMACIÓN Curso 2006-2007

PROBLEMA Menor cantidad de movimientos en un Ábaco

El ábaco es un dispositivo usado desde la antigua China y que consta de N filas sobre las que ruedan M anillos en cada una. De acuerdo a la fila en que estén, estos anillos tienen valores (v_1, \dots, v_n) . Sumando los valores de los anillos dispuestos hacia la izquierda de cada fila, se obtiene un número. De esta manera se usaba el ábaco para facilitar hacer cálculos.

La siguiente figura corresponde a dos ábacos donde los valores de los anillos en cada fila (de abajo hacia arriba) son en este caso **2**, **3**, **6** y **10**, y en cada fila hay **5** anillos. En esta figura ambas representaciones nos dan el número **25** ($2*10 + 1*3 + 1*2$ en el ábaco de la izquierda y $1*10 + 1*6 + 1*3 + 3*2$ en el ábaco de la derecha).



El problema que tenía Wang Cheng¹ era que le pesaba tener que mover muchos anillos para partiendo de la representación de un número llegar a la representación de otro número. Por ejemplo, si partimos de la figura de la izquierda (que representa un valor **25**) y queremos llegar al valor **26**, se podría lograr moviendo uno de la fila del **6** hacia la izquierda (lo que suma **6**) y luego uno de la fila del **3** y otro de la fila del **2** hacia la derecha (lo que le restaría **5**), es decir en este caso un total de **3** anillos a mover. Pero también podemos lograrlo moviendo hacia la izquierda uno del **3** y luego hacia la derecha uno del **2** y en este caso se habrían movido solo **2** anillos.

El problema que se quiere resolver es un algoritmo que partiendo de una disposición inicial del ábaco determine la menor cantidad de anillos que habría que ~~hacer~~ para obtener la representación de un determinado número.

Es decir que usted debe implementar una dll **Abaco.dll** que contenga la clase y el método siguientes

```
namespace ExamenProgramacion
{
    public class Abaco
    {
```

¹ Qin Shi Huangdi, primer emperador de china. 247-210 a. C.

```

        public static int CantidadDeMovimientos(int[] valores,
                                                int cantidadPorFila,
                                                int[] disposicion, int valorAObtener)
        {
            ...
        }
    }
}

```

Descripción de los parámetros del método.

int[] valores: Valores de los anillos para cada fila del ábaco.

int cantidadPorFila: Cantidad de anillos por fila. Recuerde que todas las filas tienen igual cantidad de anillos.

int[] disposicion: Disposición inicial del ábaco. El valor del elemento **disposición[i]** representa la cantidad de anillos de la fila **i** que están a la izquierda en la configuración de partida que se recibe del ábaco.

int valorAObtener: Número que se desea obtener en el ábaco.

Valor de Retorno:

El método debe retornar como valor un entero no negativo que represente la menor cantidad de anillos que necesitan moverse para alcanzar una representación del número especificado en el parámetro **valorAObtener**. Debe devolverse el valor **-1** en caso que sea imposible alcanzar una representación para dicho valor.

Por ejemplo si se quiere obtener la representación del **26** a partir del ábaco de la Figura anterior izquierda, entonces habría que llamar al método con los parámetros

```
CantidadDeMovimientos(new int[] { 2, 3, 6, 10 }, 5, new int[] { 1, 1, 0, 2 }, 26);
```

y el método deberá retornar **2**.