

# Examen Final de Programación

## Curso 2013-2014

### La encrucijada del Caminante

**NOTA:** Si usted está leyendo este documento sin haber extraído el compactado que descargó del sitio, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios **no se guarden**. Si usted comete este error y entrega una solución vacía, **no tendrá oportunidad de reclamar**.

Un caminante debe atravesar un río para continuar su viaje, para poder hacerlo debe auxiliarse de un grupo de cajas que se encuentra en la orilla tirando algunas de estas al río para caminar sobre ellas. Cada caja tiene una altura (valor entero mayor que cero). Para cruzar el río basta con poner 2 (siempre deben ser al menos 2 cajas) o más cajas en el río. Pero las cajas tienen que ponerse en el río siguiendo las restricciones que se explican a continuación:

1. Las cajas solo pueden ponerse en el río respetando el orden en que se encuentran en la orilla, considere el conjunto de cajas representado por un array `int[]` donde cada índice indica el orden de las cajas. El valor de cada elemento del array indica la altura de la caja correspondiente.
2. La segunda restricción es que dos cajas consecutivas colocadas en el río no pueden diferenciarse en altura en más de una cantidad `H`. Aunque el caminante siempre va a poder saltar desde la orilla a una caja y desde una caja a la orilla sin importar la altura de la caja.

La Figura 1 muestra un ejemplo de todas las combinaciones posibles (según la primera restricción siempre colocando al menos dos cajas y respetando el orden en que se encuentran en el array) utilizando de inicio cuatro cajas cuyas alturas distribuidas en el array `{5, 7, 3, 5}` respectivamente y siendo en este caso `2` el número `H` máximo posible de diferencia entre las alturas.

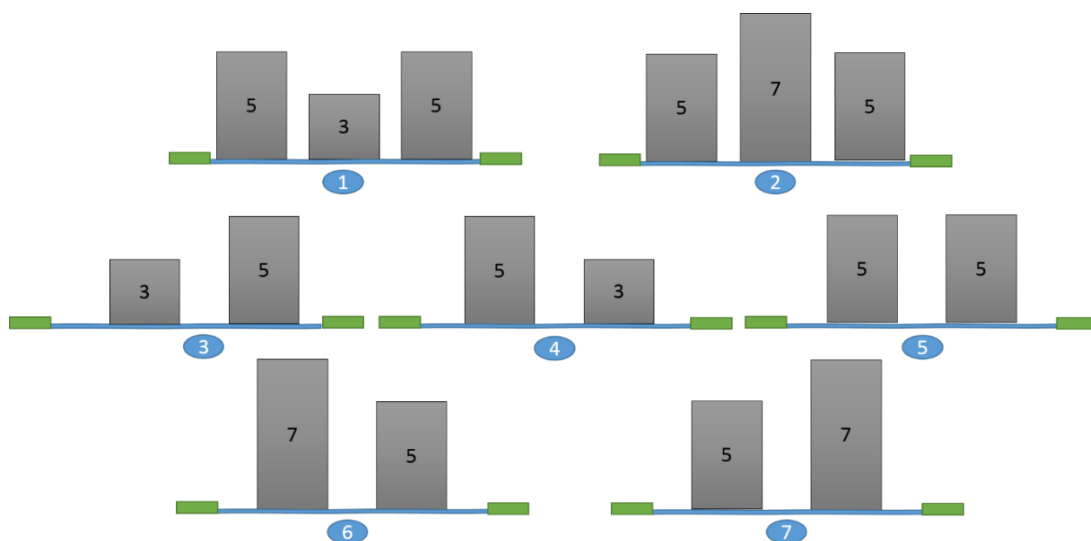


Figura 1

Como se observa en la Figura 1 en este caso son siete las posibles combinaciones de cajas que el caminante puede utilizar para cruzar el río.

Para el array **{1, 3, 5}**, si **H** fuese igual a **2** la Figura 2 muestra todas las combinaciones posibles (tres en este caso).

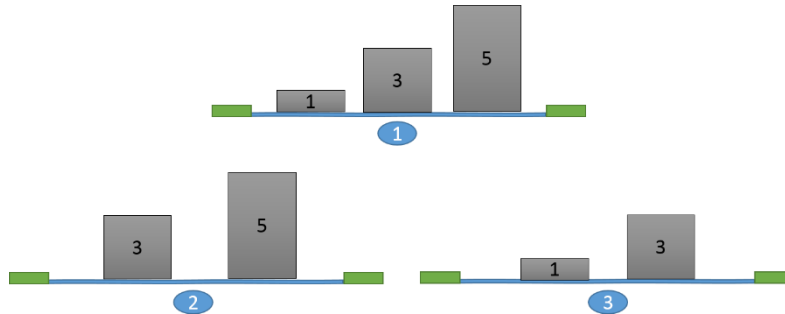


Figura 2

Pero por ejemplo si el valor de **H** fuera **1** para este mismo array **{1, 3, 5}** entonces no existiría ninguna combinación que cumpla con las restricciones del problema.

## Problema

Dados un array de enteros **cajas** con los valores de las alturas de las cajas ordenadas de acuerdo a su disposición inicial en el río y un entero **alturaMax** con la diferencia máxima entre las alturas de dos cajas colocadas de forma consecutivas en el río. Usted debe implementar el método **CantidadCombinacionesCajas** que devuelve el número de combinaciones posibles de las cajas que se pueden utilizar para cruzar el río de una orilla a la otra cumpliendo todas las restricciones del problema planteado.

```
namespace Weboo.Examen.Final
{
    public class Caminante
    {
        /// <summary>
        /// Dada una secuencias de cajas y la diferencia posible entre alturas,
        /// determinar el número de combinaciones posibles de estas cajas.
        /// </summary>
        /// <param name="cajas">Combinación inicial de cajas</param>
        /// <param name="alturaMax">Diferencia entre las alturas de dos cajas
        /// colocadas de forma consecutivas en el río</param>
        /// <returns>El número de combinaciones posibles de las cajas</returns>
        public static int CantidadCombinacionesCajas(int[] cajas, int alturaMax)
        {
            throw new NotImplementedException("Su código debe ir aquí");
            //En lugar de disparar excepción coloque aquí su implementación
        }
    }
}
```

Listado 1. Método *CantidadCombinacionesCajas* a implementar

## Aclaraciones

- El parámetro **cajas** que se le pasara al método no tendrá valor **null** pero sí puede ser un array de longitud **0**. Cada uno de los valores del array es mayor que **0**.
- El parámetro **alturaMax** siempre tendrá un valor mayor o igual que **0**.

- Como se muestra en el ejemplo de la Figura 2, en dependencia de los valores del array y del valor de `alturaMax` la respuesta puede ser `0`.