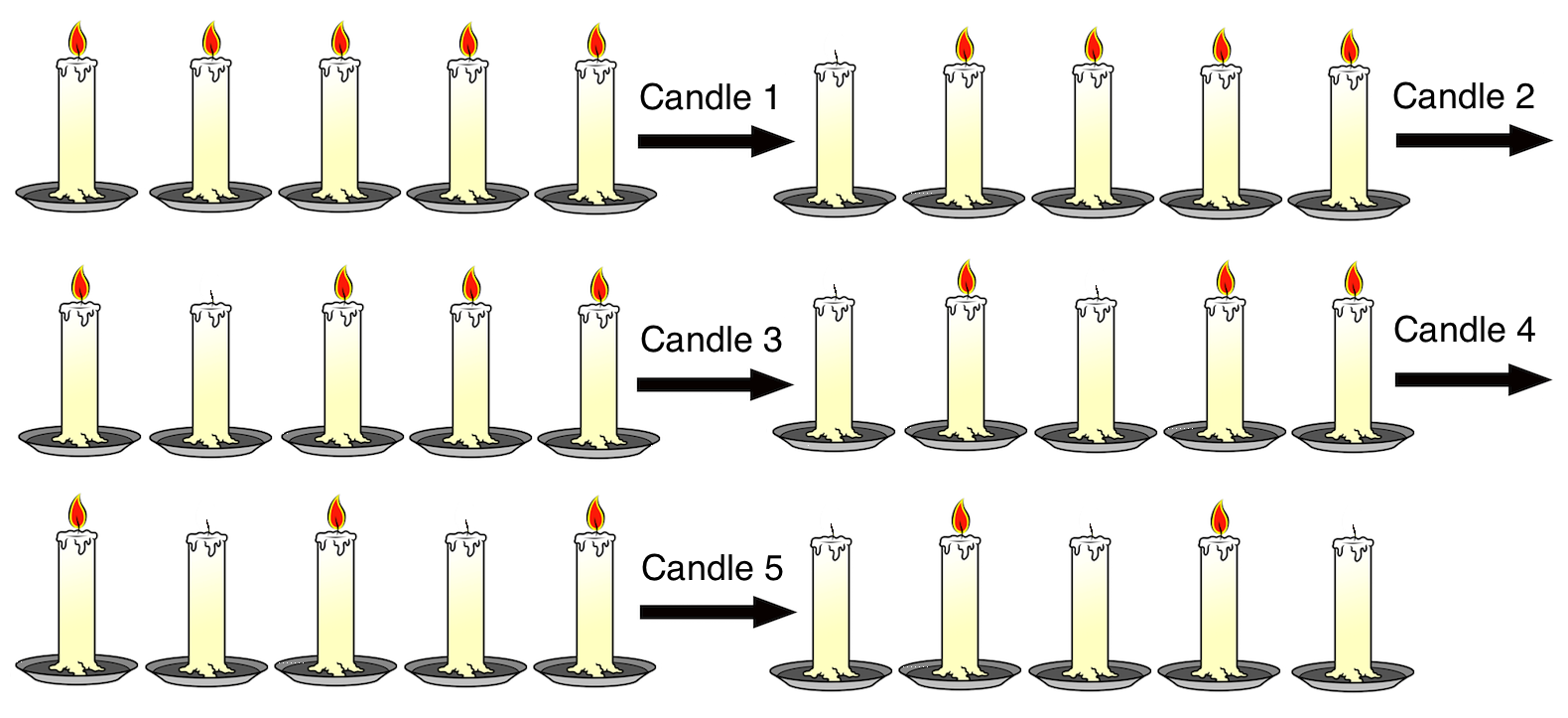
N candles are placed in a row, some of them are initially lit. For each candle from the 1stto the Nth the following algorithm is applied: if the observed candle is lit then states of this candle and all candles before it are changed to the opposite. Which candles will remain lit after applying the algorithm to all candles in the order they are placed in the line?

**Example**

For a = [1, 1, 1, 1, 1], the output should be  
switchLights(a) = [0, 1, 0, 1, 0].

Check out the image below for better understanding:



**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer a**

Initial situation - array of zeros and ones of length N, 1 means that the corresponding candle is lit.

*Constraints:*  
5 ≤ a.length ≤ 5000.

* **[output] array.integer**

Situation after applying the algorithm - array in the same format as input with the same length.

<https://codefights.com/arcade/code-arcade/well-of-integration/x3ix7CY93z2bwKDtG>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

using System.Threading;

namespace ConsoleApplication1

{

class Program

{

static int[] switchLights1(int[] a)

{

for (int i = 0; i < a.Length; i++)

{

if (a[i] == 1)

{

for (int j = i; j >= 0; j--)

{

if (a[j] == 1)

{

a[j] = 0;

}

else

{

a[j] = 1;

}

}

}

}

return a;

}

/\*es muchisimo mas rapido este algoritmo que el de arriba

lo que hago es recorrer del extremo derecho hacia el extremo izquierdo

y voy contando la cantidad de unos, lo voy almacenando a la cantidad

\* de unos en el array paralelo de frecuencias, como voy de la derecha

\* a la izquierda funciona porque en cada indice del array paralelo

\* almaceno la cantidad actual de unos, que son todos los que estan a

\* la derecha de cada indice incluido el actual

\* luego recorro el array de frecuencias para saber si en cada frec[i] es

\* par o impar, si es par no cambia nada pero si es impar cambia de 1 a 0

\* y de 0 a 1 \*/

static int[] switchLights2(int[] a)

{

int[] frec = new int[a.Length];

int unos = 0;

for (int i = a.Length - 1; i >= 0; i--)

{

if (a[i] == 1)

{

unos++;

}

frec[i] = unos;

}

for (int i = 0; i < a.Length; i++)

{

if (frec[i] % 2 != 0)

{

if (a[i] == 1)

{

a[i] = 0;

}

else

{

a[i] = 1;

}

}

}

return a;

}

static void Main(string[] args)

{

int[] a = new int[10000];

Random r = new Random();

for (int i = 0; i < a.Length; i++)

{

a[i] = r.Next(0, 100) > 50 ? 1 : 0;

}

Stopwatch stopWatch = new Stopwatch();

stopWatch.Start();

//Thread.Sleep(1000);

switchLights2(a);

stopWatch.Stop();

// Get the elapsed time as a TimeSpan value.

TimeSpan ts = stopWatch.Elapsed;

Console.WriteLine(ts.Milliseconds);

//-----------------

stopWatch = new Stopwatch();

stopWatch.Start();

//Thread.Sleep(1000);

switchLights1(a);

stopWatch.Stop();

// Get the elapsed time as a TimeSpan value.

ts = stopWatch.Elapsed;

Console.WriteLine(ts.Milliseconds);

Console.ReadLine();

}

}

}