COMPLEJIDAD COMPUTACIONAL

Fragmento de código que resuelve el problema:

**private** **void** **buscarSecuencias**() {

**int** **posicionInicio**, **posicionFin**, **longitud** = 0;

// para cada lata

**for** (**int** **i** = 0; i < **this**.latas.size(); i++) {

// si la lata es de garbanzos

**if** (**this**.latas.get(i).equals('G')) {

// comienza una secuencia de latas de garbanzos

posicionInicio = i;

// cuento todas las latas de garbanzos mientras haya latas

**do** {

longitud++;

i++;

} **while** (i < **this**.latas.size() && **this**.latas.get(i).equals('G'));

// la secuencia termina en la lata anterior

posicionFin = i - 1;

// acomodo la secuencia en el ranking

**if** (longitud > **this**.secuenciaMaslarga.getLongitud()) {

**this**.segundaSecuenciaMasLarga = **this**.secuenciaMaslarga;

**this**.secuenciaMaslarga = **new** Secuencia(longitud, posicionInicio, posicionFin);

} **else** **if** (longitud > **this**.segundaSecuenciaMasLarga.getLongitud()) {

**this**.segundaSecuenciaMasLarga = **new** Secuencia(longitud, posicionInicio, posicionFin);

}

// contador de longitud a cero

longitud = 0;

}

}

}

La complejidad computacional del algoritmo que resuelve el problema es O(n) ya que la lista de latas solo se recorre una vez