

Número de máximos

Implementar un algoritmo que verifique la siguiente especificación

```
// Cuenta el número de valores de la secuencia iguales al valor de m
function MiCount(s:seq<int>, m : int):nat
  ensures |s| == 0 ==> MiCount(s,m) == 0
  ensures |s| > 0 && m == s[0] ==> MiCount(s,m) == 1 + MiCount(s[1..],m)
  ensures |s| > 0 && m != s[0] ==> MiCount(s,m) == MiCount(s[1..],m)

method ContarMax (a : array<int>) returns (n : int, m : int)
requires a != null && a.Length > 0
ensures n == MiCount(a[..],m)
ensures forall k :: 0 <= k < a.Length ==> a[k] <= m
ensures m in a[..]
```

Requisitos de implementación.

Se pide resolver el problema sin almacenar los datos de entrada, por lo tanto cada dato solo se puede tratar una vez.

Utilizar la plantilla `PlantillaJuezNumeroCasos.cpp` que se encuentra en el campus virtual en la pestaña `laboratorio`. Como los datos de entrada no se almacenan, se resuelve el ejercicio en la función `ResuelveCaso`. Por lo tanto para este ejercicio no utilizáis la función `resolver` de la plantilla.

La entrada la leéis como si fuera por teclado, utilizando el `std::cin`. Las instrucciones

```
std::ifstream in("datos.txt");
auto cinbuf = std::cin.rdbuf(in.rdbuf()); //save old buf and redirect std::cin to casos.txt
```

del `main` redirigen la entrada de teclado a fichero. Por lo tanto los datos no se teclearán, sino que se darán en un fichero. Baja del campus virtual el fichero con los casos del ejemplo, y modifica el nombre del fichero que aparece en la instrucción `std::ifstream in("datos.txt");` al nombre que tenga tu fichero de prueba. Ejecuta el programa y las instrucciones `std::cin` leerán los datos del fichero automáticamente.

El máximo debe inicializarse al primer elemento de la entrada, ya que esta no es vacía. No se pueden utilizar los valores máximos o mínimos de un tipo de datos, ni suponer ningún valor máximo o mínimo de la entrada de datos.

La profesora ha resuelto el problema con 5 asignaciones, una instrucción condicional y una instrucción de repetición y la lectura y escritura de datos.

Dafny verifica automáticamente las dos poscondiciones correspondientes al valor de `m`. La poscondición `n == MiCount(a[..],m)` no se verifica automáticamente.

Entrada

La entrada comienza con un valor entero que indica el número de casos de prueba. Cada caso de prueba consta de dos líneas. La primera indica el número de elementos de los que hay que calcular el máximo. La segunda contiene los valores.

El número de valores de entrada es mayor que cero y menor que 20000 y el valor de los elementos se sabe que pueden almacenarse en una variable de tipo `int`.

Salida

Para cada caso de prueba se escribe en una línea el valor máximo y el número de veces que se repite separados por un carácter blanco.

Entrada de ejemplo

```
3
15
5 7 3 4 7 3 2 6 5 4 7 1 2 7 3
3
6 6 6
1
5
```

Salida de ejemplo

```
7 4
6 3
5 1
```

Autor: Isabel Pita.