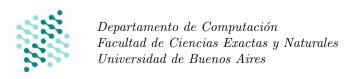
Algoritmos y Estructuras de Datos I

Primer Cuatrimestre 2020

Guía Práctica **EJERCICIOS DE TALLER**



Cambios versión 2

Ejercicio 4

- 1. Sumatoria de los elementos de una secuencia: (|s| div 2) cambió a ||s| div 2|.
- 2. Ordenar: corregimos la función mismos.

1. Debugging + Invariantes

Importar el proyecto de la clase. En CLion seleccionar File \rightarrow New CMake Project From Sources ... y seleccionar la carpeta con el nombre Labo05 que se encuentra en el archivo Labo05.zip.

En onlinegdo copiar la estructura de los archivos main.cpp, ejercicios.cpp y ejercicios.h.

Ejercicio 1. Los siguientes ejercicios contienen un código con errores junto con casos de prueba. Para los casos que fallan, utilizar el debugger para encontrar los posibles errores y arreglar el código.

- Está Ordenada: dada una secuencia de enteros indica si está ordenada ascendente o descendentemente.
- Pertenece: dada una secuencia de enteros y un elemento, indica si este pertenece o no a la secuencia.
- Fibonacci: calcula el K-esimo número de la serie de fibonacci.

Ejercicio 2. Los siguientes ejercicios contienen un código y un predicado I. Corregir el predicado para que sea un invariante del ciclo. Para cada uno, escribir Pc y Qc.

Pueden utilizar el debugger para ver el valor de las variables tanto al comenzar las iteraciones como al finalizar la ejecución del cuerpo del ciclo.

■ Suma doble: Dada una secuencia de enteros s, calcular la sumatoria del doble de los elementos que son positivos y pares.

$$I\equiv 1\leq i<|s|\wedge_L res=\sum_{k=0}^i \text{if }(s[k]\mod 2=0 \wedge s[k]>=0) \text{ then }s[k]*2 \text{ else }0 \text{ fi}$$

■ Existe Pico: Una secuencia tiene picos si en alguna posición el elemento es mayor tanto del anterior como del siguiente. Decidir si una secuencia dada (de al menos tres elementos) tiene picos.

$$I \equiv 0 \le i \le |s| \land_L res = true \leftrightarrow (\exists k : \mathbb{Z}) \ 1 \le k \le i \land_L s[k] > s[k-1] \land s[k] > s[k+1]$$

■ Todos Impares: Dada una secuencia de enteros positivos s (no vacía), chequea si todos los elementos de la secuencia son impares.

$$I \equiv 0 \le i \le |s| \land_L res = true \leftrightarrow (\forall k : \mathbb{Z})(0 \le k < i \rightarrow_L s[k] \ mod \ 2 = 1)$$

Ejercicio 3. Dado el Invariante I y un código, ver que I se cumple al principio de cada iteración y al finalizar la ejecución del cuerpo del ciclo para n = 7. ¿Qué pasa con n = 8? Corregir el codigo para que I sea un Invariante del ciclo.

■ Suma primos: Calcular la suma de todos los números primos positivos mayores a 2 hasta n (inclusive).

$$I \equiv 1 \leq i < n \wedge i \mod 2 = 1 \wedge_L suma = \sum_{k=i+2}^n \text{if } esPrimo(k) \text{ then } k \text{ else } 0 \text{ find } i \in \mathbb{N}$$

Ejercicio 4. Resolver los siguientes problemas respetando, para el ciclo principal del programa, el invariante dado.

■ Mínimo de una subsecuencia: Devuelve el índice del mínimo valor de una subsecuencia.

```
\label{eq:proc_indiceMinimoSubsec} \begin{split} & \text{proc indiceMinimoSubsec (in } s:seq\langle\mathbb{Z}\rangle, \text{ in } i,j:\mathbb{Z}, \text{ out } res:\mathbb{Z}) \quad \{ \\ & \text{Pre } \{|s|>0 \land 0 \leq i,j < |s| \land i \leq j \} \\ & \text{Post } \{i \leq res \leq j \land (\forall k:\mathbb{Z}) \ i \leq k \leq j \longrightarrow_L s[k] \geq s[res] \} \\ \} \end{split}
```

$$I \equiv i - 1 \le l \le j \land i \le res \le j \land (\forall k : \mathbb{Z}) \ l < k \le j \longrightarrow_L s[k] \ge s[res]$$

■ Sumatoria de los elementos de una secuencia: Calcula la suma de todos los elementos de una secuencia s con cantidad de elementos impares.

$$I \equiv 1 \leq i \leq \lfloor |s| \ div \ 2 \rfloor + 1 \ \wedge_L \ suma = s[\lfloor |s| \ div \ 2 \rfloor] + \sum_{k=1}^{i-1} s[\lfloor |s| \ div \ 2 \rfloor - k] + s[\lfloor |s| \ div \ 2 \rfloor + k])$$

- ¿Qué habría que cambiar en el invariante si no hubiera ninguna restricción en la cantidad de elementos de s?
- Ordenar (Opcional):Ordena ascendentemente una secuencia (no vacía) de enteros.

$$I \equiv 0 \le i \le |s| \land_L |s| = |s_0| \land mismos(s, s_0) \land_L ordenada(subseq(s, 0, i)) \land (\forall k : \mathbb{Z}) \ 0 \le k < |s| \land i > 0 \longrightarrow_L ((k < i \land s[k] \le s[i-1]) \lor (k \ge i \land s[k] \ge s[i-1]))$$

```
fun mismos(s:seq\langle\mathbb{Z}\rangle,s_0:seq\langle\mathbb{Z}\rangle): Bool =(\forall i:\mathbb{Z})\ 0\leq i<|s|\longrightarrow_L \#apariciones(s,i)=\#apariciones(s_0,i) fun ordenada(s:seq\langle\mathbb{Z}\rangle): Bool =(\forall i:\mathbb{Z})\ 0\leq i<|s|-1\longrightarrow_L s[i]\leq s[i+1] Ayuda: Utilizar la función indiceMinimoSubsec.
```