

Auxiliar 2 - Programas iterativos con invariante

Profesores: Nelson Baloian
Jeremy Barbay
Patricio Poblete

Auxiliares: Gabriel Flores, Sven Reisenegger
Elizabeth Labrada, Gabriel Norambuena
Cristóbal Muñoz

P1. Segmento de suma máxima

A Pablito le encantan los arreglos de enteros, sobre todo cuando la suma de todos sus elementos es muy grande. Cuando Pablito cumplió 8 años recibió un arreglo con números enteros, pero rápidamente se dió cuenta de que la suma total de este arreglo podía ser más grande si se quedaba sólo con una parte de este. Ayudemos a Pablito a encontrar la parte del arreglo que le sirve. Especifique cuál es el invariante del ciclo en cada caso.

- Cree una función `public void segmentoMaximo(int[] arreglo)` que resuelva el problema de encontrar el segmento con la mayor suma y que imprima en consola la suma y los índices de comienzo y fin de esta (incluidos).
- ¿Es óptimo el algoritmo creado en la parte (a)? ¿Qué optimizaciones se podrían hacer para que sea mas eficiente?

P2. Problema de partición

El ordenamiento rápido (quicksort en inglés) es un algoritmo creado por el científico británico en computación C. A. R. Hoare. La magia de este algoritmo está en el uso de la función **partición**, la cual dado un arreglo y un elemento dentro del arreglo llamado pivote, realiza dos tareas:

- Seleccionar un pivote.
- Situar el pivote en la posición que ocuparía dentro del arreglo si este estuviese ordenado.
- Sitúa todos los elementos menores o iguales que el pivote a la izquierda, y todos los elementos mayores que el pivote a la derecha de este.

Existen varios algoritmos para realizar la partición de un arreglo, unos de lo más conocidos son la partición de Hoare y la partición de Lomuto.

Partición de Hoare Si partimos con el arreglo $\{4, 5, 2, 1\}$ el arreglo variaría de la siguiente forma:

- $\{4, 1, 2, 5\}$
- $\{4, 1, 2, 5\}$
- $\{2, 1, 4, 5\}$

Partición de Lomuto Si partimos con el arreglo $\{4, 5, 2, 1\}$, el arreglo variaría de la siguiente forma:

- $\{4, 5, 2, 1\}$
- $\{4, 2, 5, 1\}$
- $\{4, 2, 1, 5\}$
- $\{1, 2, 4, 5\}$

(a) Implemente el algoritmo de la partición de Hoare con una función:

```
int HoarePartition(int [] x,int ip, int iu)
```

(b) Implemente el algoritmo de la partición de Lomuto con una función:

```
int LomutoPartition(int [] x,int ip, int iu)
```

(c) Especifique cuál es el invariante del ciclo en cada caso.

Los parámetros ip y iu son la posición inicial y final del arreglo que quiero particionar.