



Ejercicio 1. Especificar e implementar los siguientes problemas:

1. Dada una secuencia de enteros, encontrar el índice donde se encuentra el k -ésimo menor. Se define k -ésimo menor como el valor en la posición $k - 1$ si el vector estuviese ordenado de menor a mayor.
2. Dada una secuencia de enteros, retornar el índice del vector tal que la suma de los elementos a la izquierda es igual a la suma de elementos a la derecha. Si no existe tal índice, retornar el -1.

Ejercicio 2. *Compresión de secuencias:* [★] Escribir un algoritmo para el siguiente problema. Dada una secuencia de enteros retornar una secuencia de secuencias de 2 elementos $\langle begin, length \rangle$ tales que $begin$ representa el principio de la subsecuencia de elementos consecutivos y $length$ su longitud.

Por ejemplo:

- $\langle 1, 2, 3, 4, 5 \rangle$ retorna $\langle \langle 1, 5 \rangle \rangle$
- $\langle 1, 2, 3, 10, 11 \rangle$ retorna $\langle \langle 1, 3 \rangle, \langle 10, 1 \rangle \rangle$
- $\langle 1, 2, 3, 100, 102 \rangle$ retorna $\langle \langle 1, 3 \rangle, \langle 100, 1 \rangle, \langle 102, 1 \rangle \rangle$
- $\langle 5, 4, 3, 2, 1 \rangle$ retorna $\langle \langle 5, 1 \rangle, \langle 4, 1 \rangle, \langle 3, 1 \rangle, \langle 2, 1 \rangle, \langle 1, 1 \rangle \rangle$

¿Es posible a partir de la secuencia de secuencias obtenida reconstruir la secuencia original?

Ejercicio 3. ★ Dados dos strings, escribir un algoritmo que:

1. decida si el primero es una subsecuencia del segundo.
2. encuentre la subsecuencia de strings común más larga, es decir, la secuencia que es subsecuencia de ambas y que es de mayor longitud.

Ejercicio 4. Dado un entero L y un string de longitud mayor a L , escribir un algoritmo que encuentre el substring de longitud L que es el primero en orden alfabético entre todos los substrings de longitud L .

Ejercicio 5. ★ Escribir un algoritmo que dado un String que representa una frase retorne el orden inverso de las palabras en el string. Ejemplo: Dado "Hola#Mundo#Maravilloso" retorna "Maravilloso#Mundo#Hola".

Ejercicio 6. Escribir un algoritmo que dado una secuencia de Strings retorne el resultado de justificar las líneas:

Ejemplo:

```
"justifying#lines#by#####"  
"inserting#extra#blanks#is##"  
"one#task#of#a#text#editor.#"
```

Se convierte en:

```
"justifying#####lines#####by"  
"inserting#extra##blanks##is"  
"one##task#of#a#text#editor."
```

Ejercicio 7. ★ Consideremos el siguiente programa de ordenamiento, llamado ordenamiento por burbujeo (*bubble sort*):

```
for( int i = 0; i < a.size()-1; i++ ) {  
    for ( int j = 0; j < a.size()-1; j++ ) {  
        if( a[j] > a[j+1] ) {  
            swap(a, j, j+1);  
        }  
    }  
}
```

- a) Describir con palabras qué hace este programa.

- b) Proponer un invariante y variante para el ciclo principal.
- c) Proponer un invariante y variante para el ciclo interno.
- d) ¿Cuántas veces se ejecuta el swap del ciclo interior como máximo (i.e. en el peor caso)?

Ejercicio 8. ★ Ordenar los siguientes vectores utilizando los algoritmos de *ordenamiento por inserción*, *ordenamiento por selección* y *ordenamiento por burbujeo*. Para cada uno, indicar cuál (o cuáles) de los algoritmos utiliza una menor cantidad de operaciones que los demás:

- | | |
|------------------------------------|--|
| 1. $\langle 1, 2, 3, 4, 5 \rangle$ | 4. $\langle 1, 1, 1, 2, 2, 2 \rangle$ |
| 2. $\langle 5, 4, 3, 2, 1 \rangle$ | 5. $\langle 1, 2, 1, 2, 1, 2, 1, 2 \rangle$ |
| 3. $\langle 1, 3, 5, 2, 4 \rangle$ | 6. $\langle 1, 10, 50, 30, 25, 4, 6 \rangle$ |

Ejercicio 9. Escribir un algoritmo para resolver los siguientes problemas:

- Dado un vector de enteros desordenado, contar la cantidad de veces que aparece el número 0.
- Dado un vector de enteros desordenado, encontrar el número que más veces aparece en el vector.
- Dado un vector de enteros desordenado, calcular la diferencia entre el mínimo elemento del vector y el máximo elemento del vector.
- Para cada uno de los incisos anteriores, ¿Cómo podemos reducir la cantidad de iteraciones que realiza el algoritmo si el vector está ordenado?
- Resolver los tres primeros incisos para el vector $\langle 3, 1, -2, 0, , 2, -2, -2, -2, 3, 10, 0, 4 \rangle$. Si tenemos que resolver los tres ejercicios de manera consecutiva, ¿Es conveniente ordenar primero el vector?

Ejercicio 10. ★ Escribir algoritmos que resuelvan cada uno de los siguientes problemas:

- Dado un vector cuyos elementos son todos cero o uno, calcular la suma de los elementos del vector.
- Resolver el mismo problema que en el inciso anterior si se sabe que el vector está ordenado.
- Resolver el mismo problema si se sabe que el vector está ordenado, y que en lugar de cero y uno, los posibles elementos del vector son 15 y 22.

Ejercicio 11. Escribir un algoritmo que implemente este problema:

```

proc reconstruye (in a: seq(Z), out b: seq(Z)) {
  Pre { $|a| = \sum_{i=0}^{|a|-1} a[i]$ }
  Post { $|a| = |b| \wedge \text{ordenado}(b) \wedge (\forall i : \mathbb{Z})(0 \leq i < |b| \rightarrow_L 0 \leq b[i] < |a|) \wedge (\forall j : \mathbb{Z})(0 \leq j < |a| \rightarrow_L a[j] = \#apariciones(b, j))$ }
}

```

Ejercicio 12. Dar un algoritmo que resuelva este problema:

```

proc dosMitades (inout a: seq(Z)) {
  Pre { $|a| \geq 2 \wedge (\exists i : \mathbb{Z})(0 \leq i < |a| \wedge \text{ordenado}(\text{subseq}(a, 0, i)) \wedge \text{ordenado}(\text{subseq}(a, i, |a|)))$ }
  Post { $\text{ordenado}(a)$ }
}

```

Ejercicio 13. Dada una secuencia ordenada, un valor n y un valor k , retornar la secuencia con los k valores más cercanos a n .

Ejercicio 14. Dado el siguiente programa para ordenar un vector de enteros entre los índices `left` y `right`:

```
void sort(vector<int> &arr, int left, int right) {
    int i = left, j = right;
    int tmp;
    int pivot = arr[(left + right) / 2];

    /* partition */
    while (i <= j) {
        while (arr[i] < pivot) {
            i++;
        }
        while (arr[j] > pivot) {
            j--;
        }
        if (i <= j) {
            tmp = arr[i];
            arr[i] = arr[j];
            arr[j] = tmp;
            i++;
            j--;
        }
    };

    /* recursion */
    if (left < j) {
        sort(arr, left, j);
    }
    if (i < right) {
        sort(arr, i, right);
    }
}
```

1. ¿Cuál debería ser la especificación de este programa?
2. Dado un vector ya ordenado de menor a mayor, ¿cuántas iteraciones se ejecutan?
3. Se desea utilizar este programa en un contexto en el cual la subsecuencia inicial del mismo está ordenada. Optimizar el programa para que aprovechar este contexto.

Ejercicio 15. ★ Dada una secuencia de enteros, escribir un algoritmo que retorne cuántos elementos deben ser intercambiados para que esté ordenada.

Ejercicio 16. Escribir un algoritmo que, dada una matriz M de $n \times m$ ordenada de forma creciente y un elemento perteneciente a la misma, encuentre en qué fila y columna se encuentra.