

# TP 1

Juan Ignacio Sánchez Vietto - Amaya Ruth Syed

*Dpto de Computación - FCEyN*

(Dated: 9 de junio de 2014)

## I. EJERCICIO 1

### A. Ejecución

1. Abrir los documentos TP1\_with\_comments.py and runner.py.
2. Correr las funciones definidas en TP1\_with\_comments.py con runner.py como sigue:

```
# -*- coding: utf-8 -*-
```

```
import TP1_with_comments as tp
```

```
N = 1000 # cantidad de puntos de la lista de puntos aleatorios.
```

```
A = tp.abrirarchivo_txt() # cargamos un archivo de texto
```

```
B = tp.randomlist(N) # creamos una lista aleatoria de puntos
```

```
import TP1_with_comments as tp
```

```
N = 1000 # cantidad de puntos de la lista de puntos aleatorios.
```

```
A = tp.abrirarchivo_txt() # cargamos un archivo de texto
```

```
B = tp.randomlist(N) # creamos una lista aleatoria de puntos
```

```
B_ord = tp.mergesort(B) # ordenamos la lista
```

```
FB = tp.fuerzabruta(B_ord) # corremos fuerzabruta.
```

```
IG = tp.inteligente(B_ord) # corremos inteligente
```

## II. EJERCICIO 2

### A. Especificación

1.6 **esPrefijo**  $([a, b : \mathbb{C}_h]) = res : \mathbb{B} \{$

**asegura:**  $res = (|a| \geq |b|) \wedge (\forall i : \mathbb{Z})(0 \leq i < |b| \rightarrow a[i] = b[i])$   
 $\}$

**CantAp**  $([list, sec : \mathbb{C}_h]) = [x : \mathbb{Z}_h] \{$

**asegura:**  $x = \sum_{i_0=0}^{|list|-1} \beta((\forall i : \mathbb{Z})(0 \leq i < |b| \wedge a[i_0 + i] = b[i]) \wedge (\forall i_1 : \mathbb{Z})(\exists i : \mathbb{Z})(i_0 - |b| < i_1 < i_0 \wedge 0 \leq i < |b| \wedge a[i_1 + i] \neq b[i]))$   
 $\}$

**QuitarApariciones**  $([list, sec : \mathbb{C}_h]) = [res : \mathbb{C}_h] \{$

**asegura:**  $(\forall c : \mathbb{C}_h)(\text{CantAp}([c], res) = \text{CantAp}([c], list) - \text{CantAp}(sec, list) * \text{CantAp}([c], sec)) \wedge$   
 $(\forall k, l : \mathbb{Z})(0 \leq k < l < |res| \rightarrow (\exists m, n : \mathbb{Z})(0 \leq m < n < |list| \rightarrow list[m] = res[k] \wedge list[n] = res[l]))$   
 $\}$

### B. Implementación

def **esPrefijo**  $(a, b):$

```

    i = 0
    while i < len(b) and a[i] == b[i]:
        i += 1
    if i == len(b):
        return True
    else:
        return False

```

def **QuitarApariciones**  $(A, sec):$

```

    i = 0
    res = ""
    while i < len(A):
        if not esPrefijo(A[i:], sec):
            res += A[i]
            i += 1

```

```

    else:
        i+ = len(sec)
return res

```

### C. Demostración de correctitud del programa

$$1.667 \ P_c = \{i = 0 \wedge res = [] \wedge list = list_0 \wedge sec = sec_0\}$$

$$I = \{list_i = \text{mantenerhasta}(list, i) \wedge i = i_0 \wedge res = res_0 \wedge list = list_0 \wedge sec = sec_0 \wedge (0 \leq i < |list| \wedge$$

$$(\forall c : \mathbb{C}_h)(\text{CantAp}([c], res) = \text{CantAp}([c], list_i) - \text{CantAp}(sec, list_i) * \text{CantAp}([c], sec)) \wedge (\forall k, l :$$

$$\mathbb{Z})(0 \leq k < l < |res| \rightarrow (\exists m, n : \mathbb{Z})(0 \leq m < n < |list_i| \rightarrow list_i[m] = res[k] \wedge list_i[n] = res[l]))\}$$

$$B = \{i < |list|\}$$

$$f_v = |list| - i$$

$$1) \ P_c \rightarrow I$$

$$2) \ \{B \wedge I\} \langle cuerpo \rangle \text{ Vale } \{I\}$$