

## Taller 2 - Programación Dinámica

David Gutierrez Alarcon,  
Julian Andres Carrillo Chiquisa,  
David Alejandro Castillo Chiquiza

24 de septiembre de 2021

### **Resumen**

En este documento se presenta el análisis de los dos problemas planteados y sus soluciones mediante el uso de el método de Dividir y Vencer

# Análisis y Diseño del Problema

Análisis

Diseño

## Algoritmos

Evidente recursivo

Pseudocódigo

---

**Algorithm 1** EvidenteRecursivo

---

```
1: procedure EVIDENTERECURSIVO( $X, Y, Z, A, Q$ )
2:   if  $A == Q$  then
3:      $s \leftarrow \text{True}$ 
4:     return  $A$ 
5:   end if
6:   if  $s == \text{False}$  then
7:     if  $(\text{longitud}(X) == 0) \wedge (\text{longitud}(Y) == 0)$  then
8:       return  $A$ 
9:     end if
10:    if  $(\text{longitud}(X) > 0) \vee (\text{longitud}(Y) > 0)$  then
11:      if  $\text{longitud}(X) > 0$  then
12:        if  $X[0] == Z[0]$  then
13:           $\text{EvidenteRecursivo}(X[1:], Y, Z[1:], A + X[0], Q)$ 
14:        end if
15:      end if
16:      if  $\text{longitud}(Y) > 0$  then
17:        if  $Y[0] == Z[0]$  then
18:           $\text{EvidenteRecursivo}(X, Y[1:], Z[1:], Z + Y[0], Q)$ 
19:        end if
20:      end if
21:    end if
22:  end if
23:  return  $A$ 
24: end procedure
```

---

Complejidad

Invariante

Notas de Implementación

Memorizado

Pseudocódigo

---

**Algorithm 2** Memorizado

---

```
1: procedure MEMORIZADO( $X, Y, Z, M, N, C$ )
2:   if  $C[M][N] \neq \text{""}$  then
3:     return  $C[M][N]$ 
4:   end if
5:   if  $\text{longitud}(z) == 0$  then
6:      $sePudo = \text{True}$ 
7:     return  $C[M][N]$ 
8:   end if
9:   if  $\text{longitud}(X) > 0 \vee \text{longitud}(Y) > 0$  then
10:    if  $\text{longitud}(X) > 0$  then
11:      if  $X[0] == Z[0] \wedge sePudo == \text{False}$  then
12:         $C[M][N] = \text{str}(\text{Memorizado}(X[1:], Y, Z[1:], M - 1, N, C)) +$ 
13:         $X[0]$ 
14:      end if
15:    end if
16:    if  $\text{longitud}(Y) > 0$  then
17:      if  $Y[0] == Z[0] \vee sePudo == \text{False}$  then
18:         $C[M][N] = \text{str}(\text{Memorizado}(X, Y[1:], Z[1:], M, N - 1, C)) +$ 
19:         $Y[0]$ 
20:      end if
21:    end if
22:    return  $C[M][N]$ 
23: end procedure
```

---

Complejidad

Invariante

Notas de Implementación

Bottom-up

Pseudocódigo

Complejidad

Invariante

Notas de Implementación

## Comparación