

## 1 Ecuaciones de recurrencia

1. Soluciones las siguientes ecuaciones de recurrencia y determine el orden de complejidad, para las ecuaciones:

I Proponer una cota usando arboles de recurrencia.

II Demostrar la cota propuesta usando el método de sustitución.

III **De ser posible** compruebe su respuesta usando el método maestro.

.....

(a)  $T(n) = T(n-1) + n$ ,  $T(0) = 1$ ,  $T(1) = 5$ .

(b)  $T(n) = T(n-2) + n^2$ ,  $T(0) = 1$ ,  $T(1) = 5$ .

(c)  $T(n-2) = 2T(n) - T(n-1)$ ,  $T(0) = 1$ ,  $T(1) = 5$ .

(d)  $T(n) = 3T(\frac{n}{2}) + n$ ,  $T(0) = 1$ .

(e)  $T(n) = 7T(\frac{n}{2}) + 20n^2$ .

(f)  $T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + T(\frac{n}{8}) + n$ .

(g)  $T(n) = 4T(\frac{n}{2}) + n^2\sqrt{n}$ .

## 2 Programación dinámica

1. Ejercicios propuestos de LeetCode, para estos ejercicios se deben hacer tres implementaciones y comparar las complejidades temporales de cada una:

I Fuerza bruta

II Dividir y conquistar

III Programación dinámica

(a) [62. Unique Paths](#)

(b) [91. Decode Ways](#)

(c) [152. Maximum Product Subarray](#)

(d) [322. Coin Change](#)

(e) [329. Longest Increasing Path in a Matrix](#)

(f) [494. Target Sum](#)

(g) [1143. Longest Common Subsequence](#)