

Para cada ejercicio se deben hacer dos implementaciones y comparar las complejidades temporales de cada una:

1. Dividir y conquistar
2. Programación dinámica
1. **Subsecuencia alternante** Dada una cadena/arreglo de números este problema consiste en encontrar la **longitud** de la subsecuencia alternante mas larga que esta contenida en la arreglo dado. Por ejemplo, para $A = [7, 10, 6, 4, 5, 7, 0, 2, 5]$ la longitud es 6, la subsecuencia es $[7, 10, 6, 7, 0, 5]$. La complejidad temporal esperada es $\mathcal{O}(n^2)$.
2. **Subsecuencia bitonica** Dada una cadena/arreglo de números este problema consiste en encontrar la subsecuencias bitonicas mas larga que esta contenida en la cadena/arreglo dado. Los elementos de la subsecuencia se ordenan primero en orden creciente, luego en orden decreciente La complejidad temporal esperada es $\mathcal{O}(n^2)$.
3. **Subsecuencia palindrómica** Dada una cadena/arreglo de números este problema consiste en encontrar la longitud de la subsecuencias palindrómicas mas larga que esta contenida en la cadena/arreglo dado. Por ejemplo, 1,2,3,4,0,-1 es un subsecuencia . La complejidad temporal esperada es $\mathcal{O}(n^2)$.
4. **Edit distance** Dadas dos strings (w_1, w_2) se quiere conocer cual es el numero mínimo de operaciones, inserciones, eliminaciones y remplazo para llegar de w_1 a w_2 . Por ejemplo, si se tiene $w_1 = \text{'This'}$ y $w_2 = \text{'That'}$ el numero mínimo de operaciones es 2. Remplazar la 'i' por una 'a' y la 's' por una 't'. (**Cual es la mejor complejidad que se puede lograr?**)
5. **Knapsack** Dado N elementos, cada uno asignado con un valor V_i un peso W_i ($\forall i \in [0 \dots N - 1]$), y el tamaño de una maleta S . La tarea consiste en colocar los objetos en la maleta de tal forma que la suma de los valores asociados a ellos sea la máxima posible. Por ejemplo, si $N = 3, W = 4, V = [1, 2, 3], W = [4, 5, 1]$ la respuesta del problema debe ser 3, solo se puede llevar en la maleta el tercer elemento. Cualquier otra posibilidad supera el limite de capacidad de la maleta. (**Cual es la mejor complejidad que se puede lograr?**)
6. Se proponen los siguientes ejercicios de leetcode:
 - (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](#)