



# Greedy Dos Punteros

Jeremías Broin Luque y Eduardo Carranza Vélez



# Greedy

Un algoritmo "greedy" construye una solución al problema haciendo siempre una elección que parece ser la mejor en este momento. Un algoritmo greedy nunca retrocede sus elecciones, sino que construye directamente la solución final. Por esta razón los algoritmos greedy suelen ser muy eficientes.

La dificultad en diseñar algoritmos greedy es encontrar una estrategia que siempre produce una solución óptima al problema. Las opciones localmente óptimas en un algoritmo greedy también debería ser globalmente óptimo. Normalmente lo difícil es demostrar que un algoritmo greedy funciona.

# Problema motivador:

## **Dragons:**

Tenemos  $n$  dragones que tienen vida  $x_i$  y nos dan fuerza  $y_i \geq 0$  una vez que los matamos. Para matar un dragón requerimos que nuestra fuerza sea  $\geq x_i$ . Inicialmente empezamos con fuerza  $f$ . Decidir si podemos matar a todos los dragones.

Solución:

# Problema para practicar:

## **Remove Smallest**

<https://codeforces.com/contest/1399/problem/A>

# Problema 3

## Movie Festival

Dadas  $n$  películas con inicio  $x_i$  y final  $y_i$ . Decidir la máxima cantidad de películas COMPLETAS que se pueden ver. (El cambio entre películas es inmediato y solo se puede ver una película a la vez)

# Problema 4

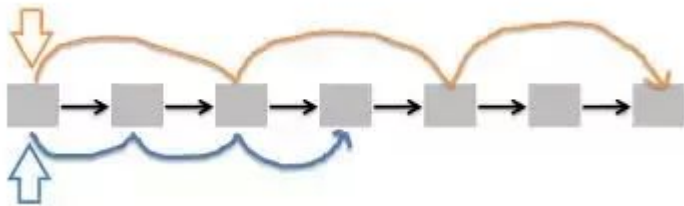
## Tasks and Deadlines

Dadas  $n$  tareas con duración  $d_i$  y deadline  $x_i$ , se deben finalizar todas las tareas. Al finalizar una tarea recibimos  $x_i - t$  puntos, donde  $t$  es el tiempo en que finalizamos la tarea. Decidir la máxima cantidad de puntos que podemos obtener eligiendo el orden de realización.

# Dos punteros

En el método de dos punteros, utilizamos dos índices para iterar sobre los valores del arreglo.

Ambos punteros se pueden mover en una sola dirección, lo que garantiza que el algoritmo funciona eficientemente. Específicamente, los punteros se moverán en total  $O(n)$  posiciones.





# Problema motivador:

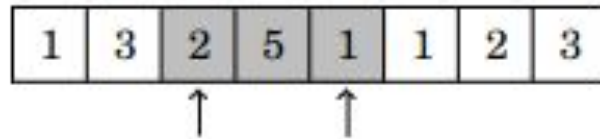
## **Subarray sum**

Dado un arreglo de enteros positivos y un número  $x$ , determinar la cantidad de subarreglos con suma  $x$ .

Solución:

## Solución:

Consideremos el siguiente array con  $x=8$ :



# Problemas para practicar:

## **Sum of two values**

<https://cses.fi/problemset/task/1640>

## **Subarray Distinct Values**

<https://cses.fi/problemset/task/2428>

¿MUCHAS GRACIAS POR VER!!!!!!!!!!!!!!111!1!

