Clase 03 - Algoritmos y complejidad

IIC1001 - Algoritmos y Sistemas Computacionales

Cristian Ruz – cruz@ing.puc.cl

Lunes 13-Marzo-2023

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Contenidos

Contenidos

Contacto

- ignaciomunoz@uc.cl Ignacio Muñoz. Ayudante 👾
  - Coordinación, logística
  - Justificación de inasistencias
  - Notas en planilla, actualizaciones
  - Todo lo que no sé donde más enviar

- cruz@ing.puc.cl Profesor 👨🏻‍🏫
  - Materia
  - Situaciones especiales

Contenidos

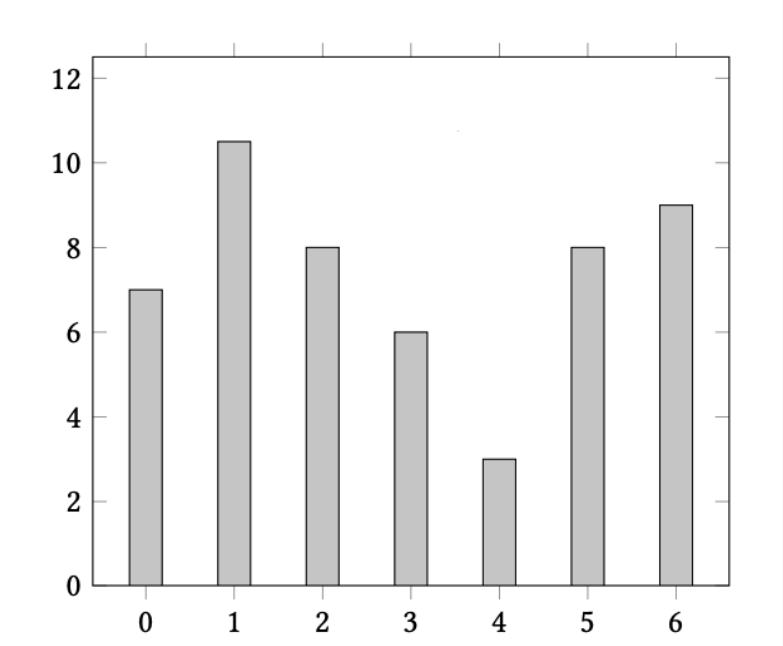
Un problema

Precios de acciones (*stock price quote*)

Secuencia de precios diarios

- Eje X: días
- Eje Y: precio del día

Pregunta: para cada día, ¿cuántos días seguidos hacia atrás el precio fue menor o igual al actual?



Un problema

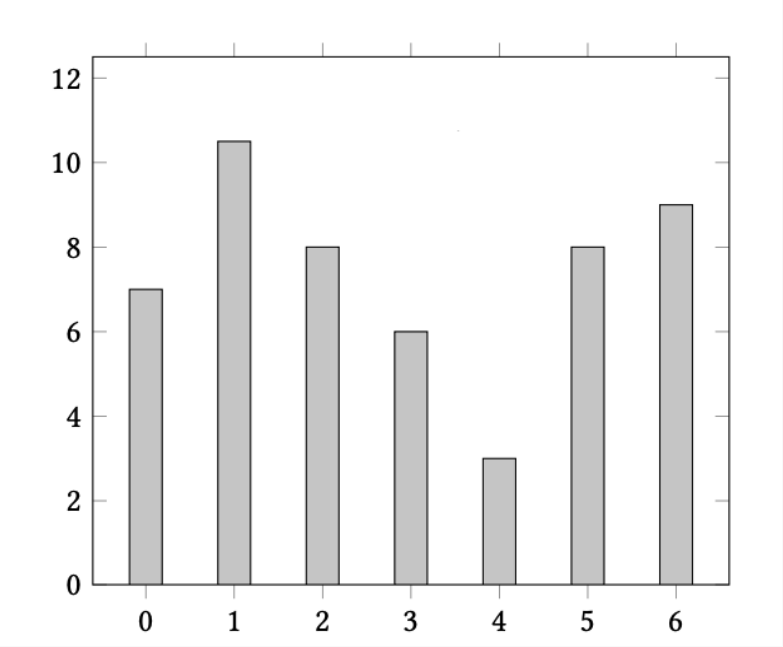
Stock Span Problem

Problema del intervalo de acciones. Dada una secuencia de precios, calcular el span (cantidad de días que el precio no ha bajado) para cada día de la secuencia.

Secuencia de precios diarios

- Eje X: días
- Eje Y: precio del día

Calcular el span para cada día.



Un problema

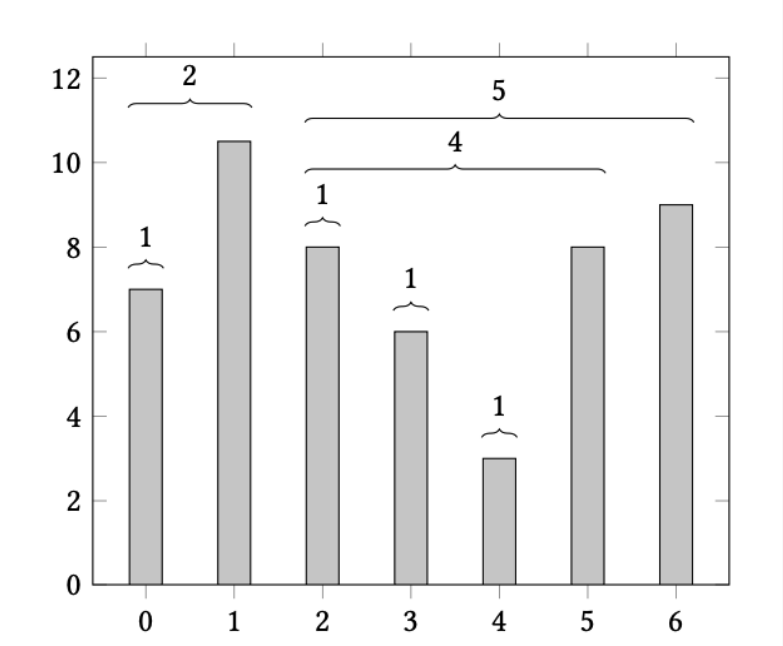
Stock Span Problem

Problema del intervalo de acciones. Dada una secuencia de precios, calcular el span (cantidad de días que el precio no ha bajado) para cada día de la secuencia.

Secuencia de precios diarios

- Eje X: días
- Eje Y: precio del día

Calcular el span para cada día.



Un problema

Stock Span Problem

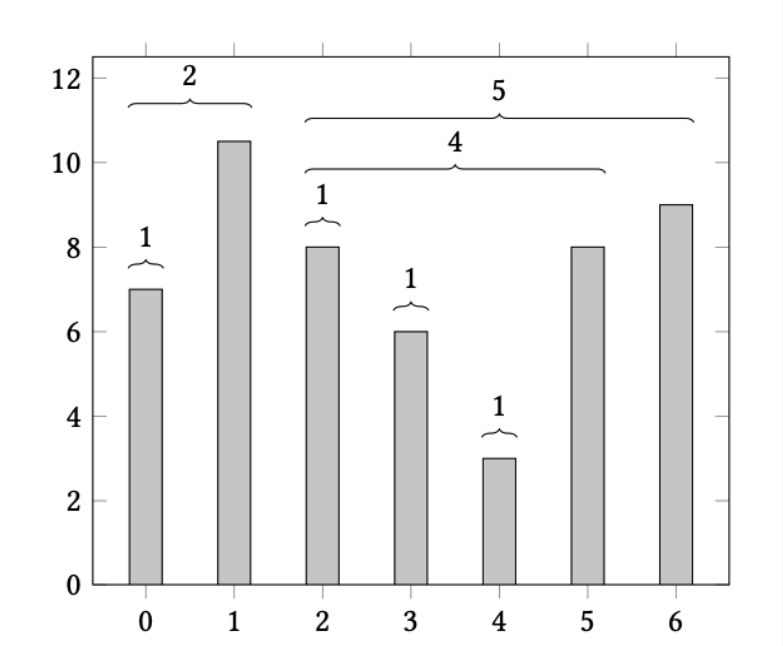
Problema del intervalo de acciones. Dada una secuencia de precios, calcular el span (cantidad de días que el precio no ha bajado) para cada día de la secuencia.

Secuencia de precios diarios

- Eje X: días
- Eje Y: precio del día

Calcular el span para cada día.

¿Cómo escribimos una solución (algoritmo)?



Un problema

Stock Span Problem

Problema del intervalo de acciones. Dada una secuencia de precios, calcular el span (cantidad de días que el precio no ha bajado) para cada día de la secuencia.

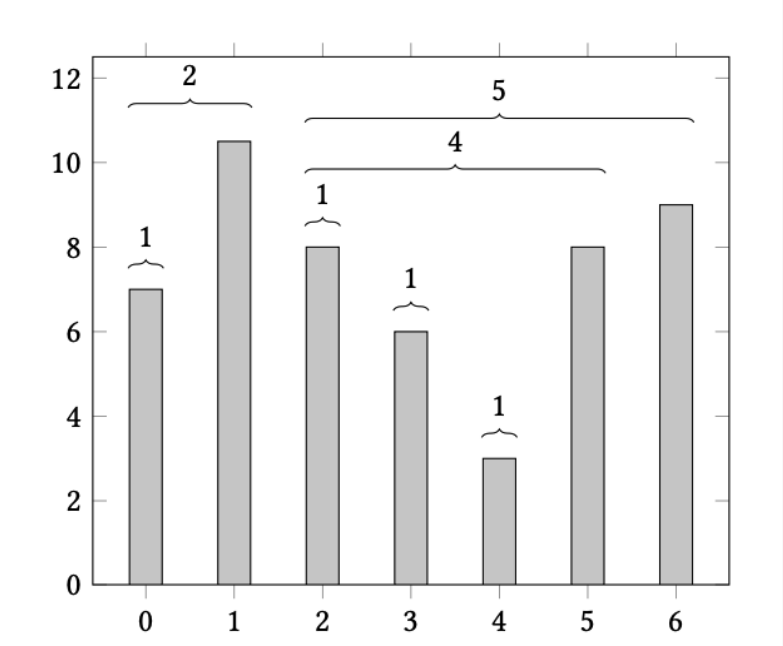
Secuencia de precios diarios

- Eje X: días
- Eje Y: precio del día

Calcular el span para cada día.

¿Cómo escribimos una solución (algoritmo)?

¿Cuál es un buen algoritmo para calcular esto?



¿Cómo se ven este algoritmo?

Calcular span de un día

1. Tomar un día m
2. Retroceder al día $m - 1$
3. Si el precio es mayor que el día m
   3.1 El span es 1
4. Si el precio es menor o igual al día m
   4.1 El span es al menos 2
   4.2 Ir al día anterior y hacer lo mismo
5. Si se acaban los días
   5.1 El span es m
6. Si nos detenemos en el día k
   6.1 El span es $m - k$
7. FIN

Calcular span de toda la serie

- Recorrer días del 1 al m
  1. Calcular span para día m

¿Cómo se ve este algoritmo?

Algorithm 1.1: A simple Stock Span algorithm.

`SimpleStockSpan(`*quotes*`)` $\rightarrow$ *spans*

Input: *quotes*, an array with n stock price quotes

Output: *spans*, an array with n stock price spans

```
1   spans ← CreateArray(n)
2   for i ← 0 to n do
3       k ← 1
4       span_end ← FALSE
5       while i − k ≥ 0 and not span_end do
6           if quotes[i − k] ≤ quotes[i] then
7               k ← k + 1
8           else
9               span_end ← TRUE
10      spans[i] ← k
11  return spans
```

¿Cuánto demora este algoritmo?

Algorithm 1.1: A simple Stock Span algorithm.

```
SimpleStockSpan(quotes) → spans
    Input: quotes, an array with n stock price quotes
    Output: spans, an array with n stock price spans
1   spans ← CreateArray(n)
2   for i ← 0 to n do
3       k ← 1
4       span_end ← FALSE
5       while i − k ≥ 0 and not span_end do
6           if quotes[i − k] ≤ quotes[i] then
7               k ← k + 1
8           else
9               span_end ← TRUE
10      spans[i] ← k
11  return spans
```

¿Cómo medimos la ejecución?

¿Tiempo (segs)?, ¿número de pasos?

¿Cuánto demora este algoritmo?

Algorithm 1.1: A simple Stock Span algorithm.

```
SimpleStockSpan(quotes) → spans
    Input: quotes, an array with n stock price quotes
    Output: spans, an array with n stock price spans
1   spans ← CreateArray(n)
2   for i ← 0 to n do
3       k ← 1
4       span_end ← FALSE
5       while i − k ≥ 0 and not span_end do
6           if quotes[i − k] ≤ quotes[i] then
7               k ← k + 1
8           else
9               span_end ← TRUE
10      spans[i] ← k
11  return spans
```

¿Cómo medimos la ejecución?

¿Tiempo (segs)?, ¿número de pasos?

Tiempo de ejecución → Número de pasos