

Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Price \$	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
Rate of Change		13	-3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7

Se desea maximizar un subarray del array rate of change, ya que es quien define las ganancias.

Tabulation

Sea cada celda la ganancia máxima hasta el día correspondiente a dicha celda. En cada día la comparación que maximiza la ganancia está dada por:

Max(Rate of Change del nuevo día, Ganancia acumulada de n días previos + Rate of Change del nuevo día)

Se parte de que el primer día no hay rate of change por lo que se empieza con máximo subarray de 13 (113-100), luego se va iterando con cada valor, $-3 < 10$ (13-3) y así se sigue realizando las comparaciones con la lógica del recuadro.

Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rate of Change	13	10	-15	20	17	1	-22	18	38	31	43	38	16	31	27	34

En otras palabras, empezar un nuevo arreglo o agregar mi nueva ganancia al existente. En base a esto, para obtener el máximo subarray se trabajan con i y j como iteradores, ambos inician en la primera posición. 'i' representará al máximo y 'j' a un temporal que validará como cambia el subarray conforme se itera la lista.

En cada nueva iteración de la lista se validará lo siguiente:

J // Temp = MAX(newElement, Temp + newElement)

Esto quiere decir que si el subarray tenía suma negativa, se empieza un nuevo array en vez de aumentar el existente. newElement solo será mayor a (Temp + newElement) con Temp negativo.

I // MAXIMO = MAX(MAXIMO, Temp)

Esto quiere decir que se verifica en cada iteración si hay que actualizar la suma nueva del subarray máximo, de no ser el caso, se mantiene el anterior.

- Para sacar los índices de inicio y fin se sigue un procedimiento sumamente similar, tomando en cuenta dos iteradores i (inicio) y j (fin) nuevos, tenemos que:

```
i = j = array[0]
```

```
//La lógica para temp y maximo y ya fue descrita
```

- para i:

Tomará el índice (pos) del elemento iterado si $\text{Array}[\text{pos}] > \text{Array}[\text{pos}] + \text{maximo}$, de lo contrario, i mantiene su posición actual

- para j:

Tomará el índice (pos) del elemento iterado si $\text{Array}[\text{pos}] > \text{Array}[\text{pos}] + \text{temp}$, de lo contrario, j mantiene su posición actual