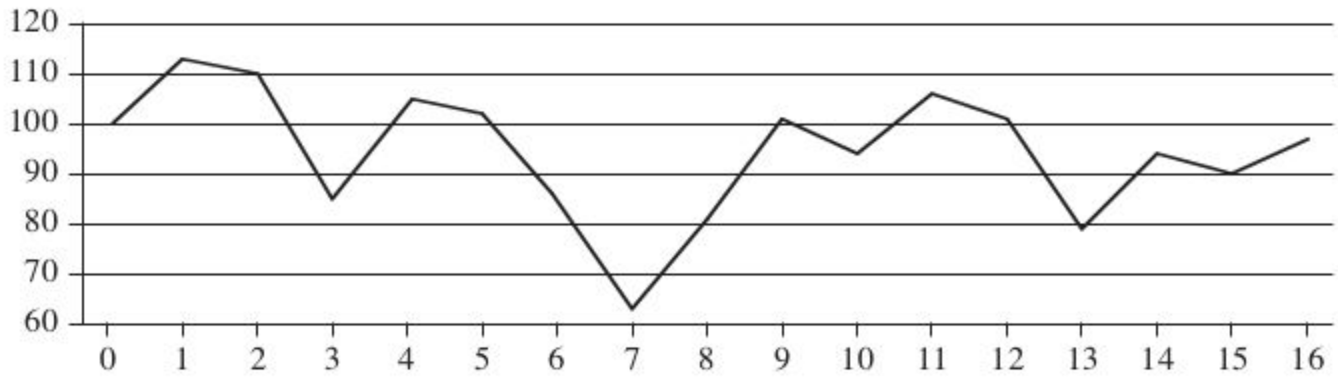
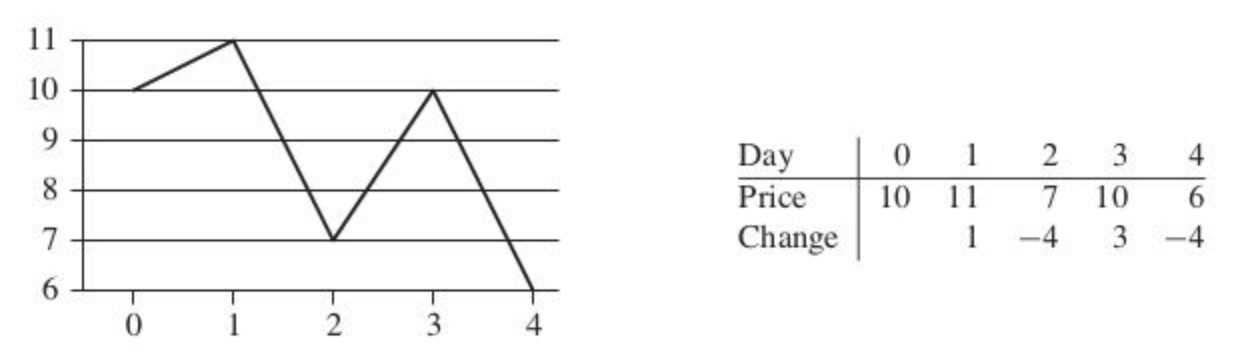
Stock prices fluctuations



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
Change		13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

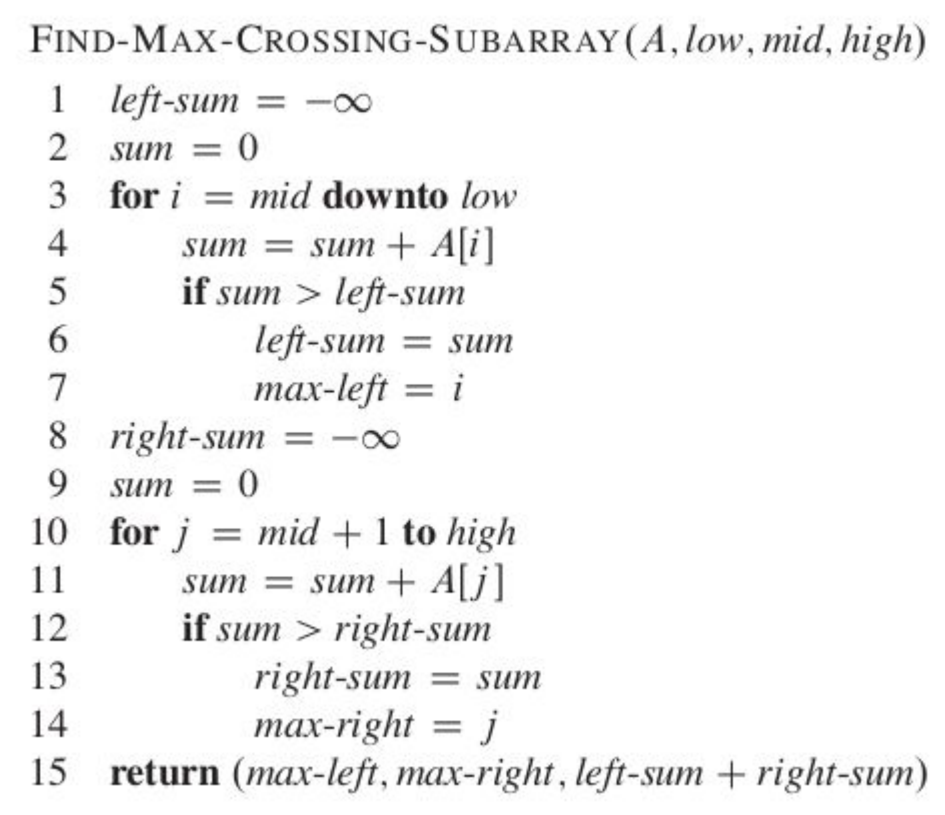
Brute force time complexity?

Stock prices fluctuations



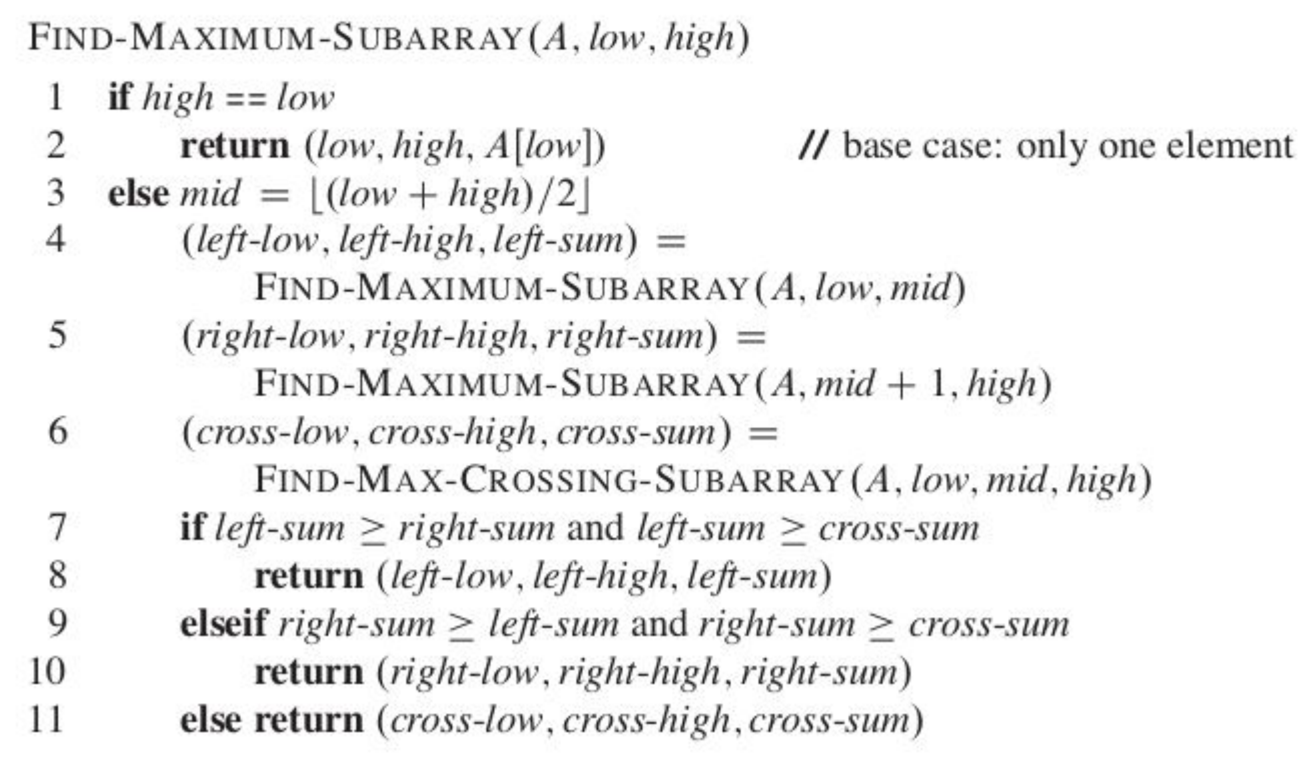
Day	0	1	2	3	4
Price	10	11	7	10	6
Change		1	−4	3	−4

Max subarray problem



```
FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
left-sum = $-\infty$
sum = 0
for $i$ = mid downto low
    sum = sum + $A[i]$
    if sum > left-sum
        left-sum = sum
        max-left = $i$
right-sum = $-\infty$
sum = 0
for $j$ = mid + 1 to high
    sum = sum + $A[j]$
    if sum > right-sum
        right-sum = sum
        max-right = $j$
return (max-left, max-right, left-sum + right-sum)
```

Max subarray problem



```
FIND-MAXIMUM-SUBARRAY(A, low, high)
if high == low
    return (low, high, A[low])              // base case: only one element
else mid = $\lfloor (low + high)/2 \rfloor$
    (left-low, left-high, left-sum) =
        FIND-MAXIMUM-SUBARRAY(A, low, mid)
    (right-low, right-high, right-sum) =
        FIND-MAXIMUM-SUBARRAY(A, mid + 1, high)
    (cross-low, cross-high, cross-sum) =
        FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
    if left-sum $\geq$ right-sum and left-sum $\geq$ cross-sum
        return (left-low, left-high, left-sum)
    elseif right-sum $\geq$ left-sum and right-sum $\geq$ cross-sum
        return (right-low, right-high, right-sum)
    else return (cross-low, cross-high, cross-sum)
```

Reference reading

Introductions to Algorithms, 3rd Edition, by Cormen, et. al.

- §4.1 The maximum subarray problem