

모두의 알고리즘 with 파이썬

문제 18 최대 수익 알고리즘

어떤 주식에 대해 특정 기간 동안의 가격 변화가 주어졌을 때, 그 주식 한 주를 한 번 사고팔아 얻을 수 있는 최대 수익을 계산하는 알고리즘을 만들어 보세요.



방법 ①: 가능한 모든 경우를 비교하기

```
# 주어진 주식 가격을 보고 얻을 수 있는 최대 수익을 구하는 알고리즘
# 입력: 주식 가격의 변화 값(리스트: prices)
# 출력: 한 주를 한 번 사고팔아 얻을 수 있는 최대 수익 값
```

```
def max_profit(prices):
    n = len(prices)
    max_profit = 0 # 최대 수익은 항상 0 이상의 값

    for i in range(0, n - 1):
        for j in range(i + 1, n):
            # i날에 사서 j날에 팔았을 때 얻을 수 있는 수익
            profit = prices[j] - prices[i]
            # 이 수익이 지금까지 최대 수익보다 크면 값을 고침
            if profit > max_profit:
                max_profit = profit

    return max_profit

stock = [10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]
print(max_profit(stock))
```

방법 ②: 한 번 반복으로 최대 수익 찾기

파는 날을 기준으로 이전 날들의 주가 중 최솟값으로 최대 수익 계산

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]

방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

최대수익 : 0
최저 주가



[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]

방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

최대수익 : 0

최저 주가



[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]



그날의 주가

그날의 주가 - 최저 주가 = -700

방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

최대수익 : 0

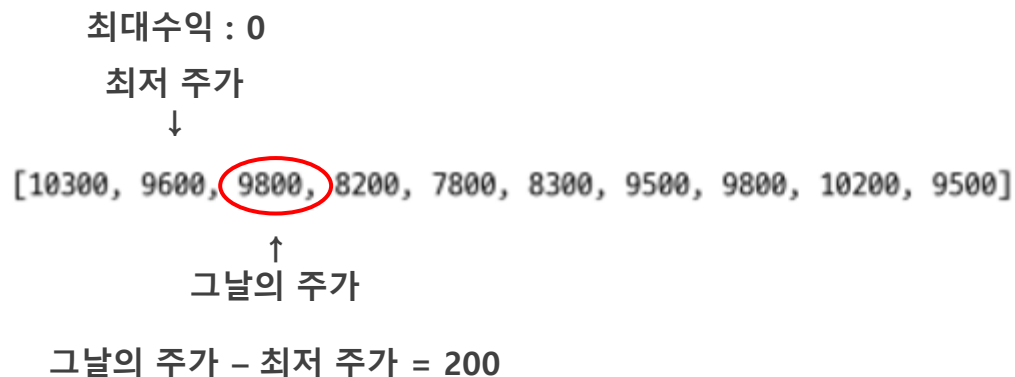
최저 주가



[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]

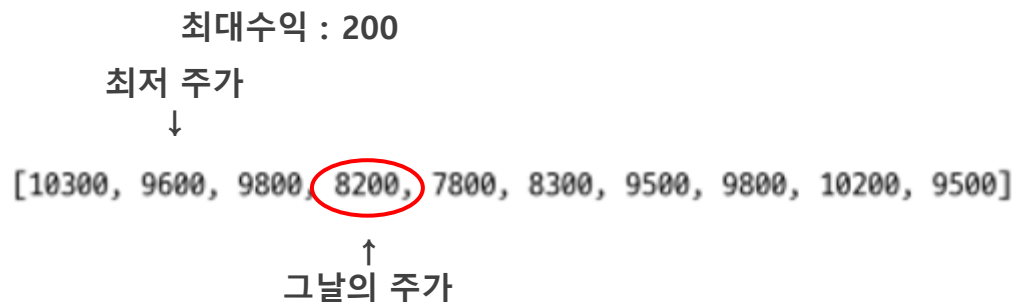
방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.



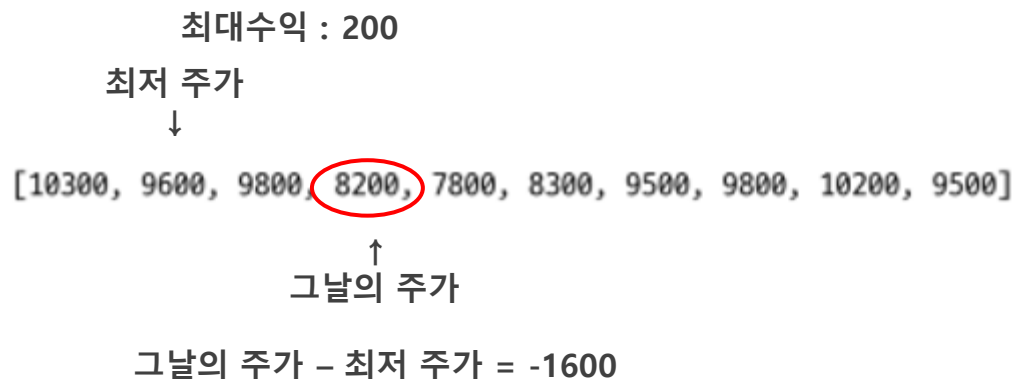
방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.



방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.



방법 ②: 한 번 반복으로 최대 수익 찾기

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

최대수익 : 200

최저 주가



[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]



그날의 주가

방법 ②: 한 번 반복으로 최대 수익 찾기

주어진 주식 가격을 보고 얻을 수 있는 최대 수익을 구하는 알고리즘
입력: 주식 가격의 변화 값(리스트: prices)
출력: 한 주를 한 번 사고팔아 얻을 수 있는 최대 수익 값

```
def max_profit(prices):  
    n = len(prices)  
    max_profit = 0      # 최대 수익은 항상 0 이상의 값  
    min_price = prices[0] # 첫째 날의 주가를 주가의 최솟값으로 기억  
    for i in range(1, n): # 1부터 n - 1까지 반복  
        # 지금까지의 최솟값에 주식을 사서 i날에 팔 때의 수익  
        profit = prices[i] - min_price  
        if profit > max_profit: # 이 수익이 지금까지 최대 수익보다 크면 값을 고침  
            max_profit = profit  
        if prices[i] < min_price: # i날 주가가 최솟값보다 작으면 값을 고침  
            min_price = prices[i]  
  
    return max_profit
```

```
stock = [10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]  
print(max_profit(stock))
```

- 1 | 최대 수익을 저장하는 변수를 만들고 0을 저장합니다.
- 2 | 지금까지의 최저 주가를 저장하는 변수를 만들고 첫째 날의 주가를 기록합니다.
- 3 | 둘째 날의 주가부터 마지막 날의 주가까지 반복합니다.
- 4 | 반복하는 동안 그날의 주가에서 최저 주가를 뺀 값이 현재 최대 수익보다 크면 최대 수익 값을 그 값으로 고칩니다.
- 5 | 그날의 주가가 최저 주가보다 낮으면 최저 주가 값을 그날의 주가로 고칩니다.
- 6 | 처리할 날이 남았으면 4번 과정으로 돌아가 반복하고, 다 마쳤으면 최대 수익에 저장된 값을 결괏값으로 돌려주고 종료합니다.

[10300, 9600, 9800, 8200, 7800, 8300, 9500, 9800, 10200, 9500]

계산복잡도

방법 ①: 가능한 모든 경우를 비교하기

$O(n^2)$

방법 ②: 한 번 반복으로 최대 수익 찾기

$O(n)$