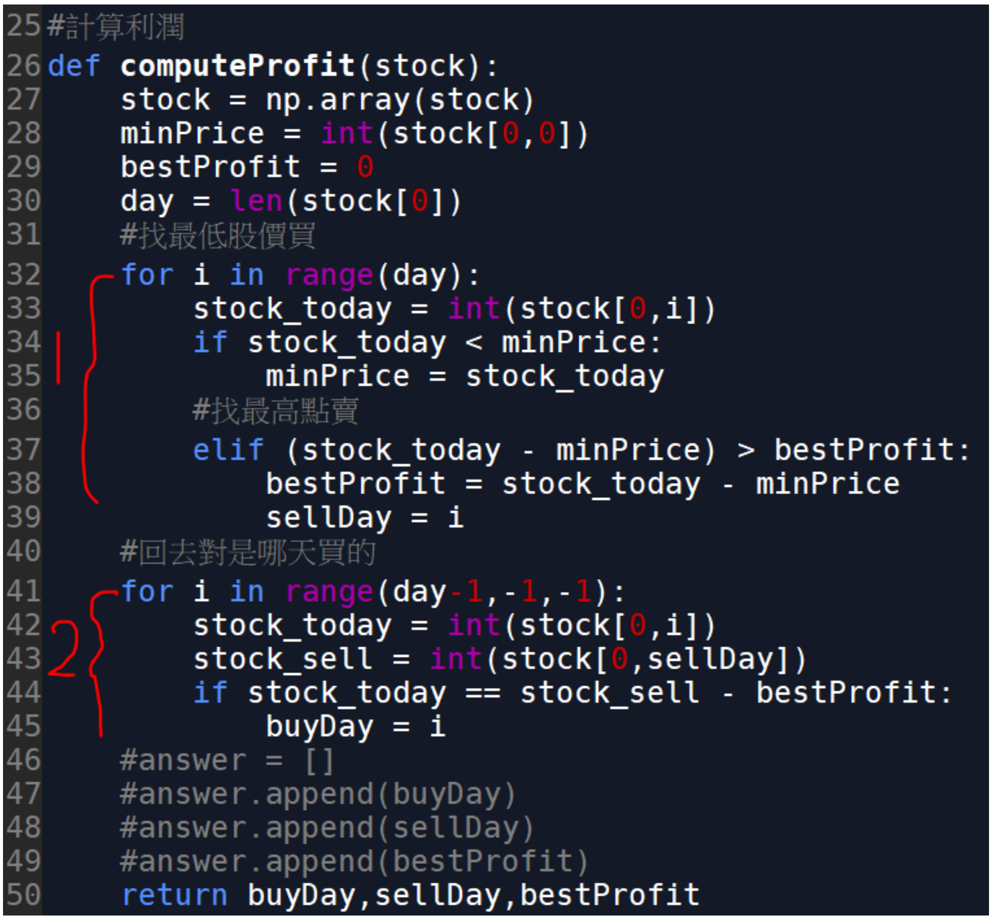
Data Structure Programming Report 4

R05546014 洪紹綺



```python
#計算利潤
def computeProfit(stock):
    stock = np.array(stock)
    minPrice = int(stock[0,0])
    bestProfit = 0
    day = len(stock[0])
    #找最低股價買
    for i in range(day):
        stock_today = int(stock[0,i])
        if stock_today < minPrice:
            minPrice = stock_today
        #找最高點賣
        elif (stock_today - minPrice) > bestProfit:
            bestProfit = stock_today - minPrice
            sellDay = i
    #回去對是哪天買的
    for i in range(day-1,-1,-1):
        stock_today = int(stock[0,i])
        stock_sell = int(stock[0,sellDay])
        if stock_today == stock_sell - bestProfit:
            buyDay = i
    #answer = []
    #answer.append(buyDay)
    #answer.append(sellDay)
    #answer.append(bestProfit)
    return buyDay,sellDay,bestProfit
```

1. 欲找出最高利潤，需先找出最低進場點，先將最低價存為第一天的股價，再依照日期順序，找出最低股價，同時依次確認每天價錢與目前的最低股價相減是否為目前的最高利潤，若為最高利潤，則記錄當下利潤及日期。程式最多需執行 n 次(n 為天數)。

2. 找出最高利潤後，套用 Backward Induction 的概念，核對今日股價與最佳利潤，找出股票是哪一天買入的。程式最多需執行 n 次(n 為天數)。