ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN

BÀI TẬP VỀ NHÀ NHÓM 3

Môn học: Phân tích và thiết kế thuật toán

Sinh viên thực hiện:
Đặng Quốc Cường
Nguyễn Đình Thiên Quang
(Nhóm 1)

Giảng viên môn học:
Nguyễn Thanh Sơn

Ngày 31 tháng 10 năm 2024

Mục lục

1 Bài 1: Hệ thống quản lý đơn hàng

1.1 Mã giả cho bài toán

```
Function TinhChiPhi(Order order):
    totalCost = 0
    For each product in order.productList:
        discountedPrice = product.price * (1 - product.discountRate)
        totalCost += discountedPrice * product.quantity

    If totalCost >= 1000000:
        shippingFee = 0
    Else:
        shippingFee = order.shippingFee

    If order.isRegularCustomer:
        totalCost = totalCost * 0.9

    totalCost += shippingFee
    Return totalCost
```

1.2 Unit Test, White Box Test và Black Box Test

Phần áp dụng Unit Test:

- Tính toán giá trị tổng chi phí (`TinhChiPhi`).
- Tính tổng chi phí từng sản phẩm sau khi áp dụng giảm giá.
- Xác định phí vận chuyển dựa trên tổng chi phí ban đầu.
- Áp dụng chiết khấu cho khách hàng thường xuyên.

Phần áp dụng White Box Test:

- Các nhánh điều kiện (nếu tổng >= 1 triệu, nếu là khách hàng thường xuyên).
- Xác định tất cả các đường đi để kiểm tra tính chính xác của các trường hợp tính phí vận chuyển và chiết khấu.

Phần áp dụng Black Box Test:

- Kiểm tra với đầu vào không có giảm giá, có giảm giá, tổng chi phí trước và sau khi giảm giá vượt quá 1 triệu.
- Các trường hợp có/không phải khách hàng thường xuyên.

2 Bài 2: Dãy con có tổng lớn nhất

2.1 Giải pháp với độ phức tạp $O(n^2)$ hoặc $O(n^3)$

```
Function MaxSubarraySum_N2(arr, n):
    maxSum = -Infinity
    For i = 1 to n:
        currentSum = 0
        For j = i to n:
            currentSum += arr[j]
            If currentSum > maxSum:
                maxSum = currentSum
    Return maxSum
```

2.2 Giải pháp tối ưu với độ phức tạp $O(n)$ (Kadane's Algorithm)

```
Function MaxSubarraySum_Kadane(arr, n):
    maxSum = arr[0]
    currentSum = arr[0]
    For i = 1 to n:
        currentSum = max(arr[i], currentSum + arr[i])
        maxSum = max(maxSum, currentSum)
    Return maxSum
```

2.3 Trình sinh test và so sánh giữa "code trâu" và "code full"

```
Function GenerateTestCases():
    testCases = []
    For i = 1 to numTests:
        arr = RandomArrayOfIntegers(size, minValue, maxValue)
        testCases.append(arr)
    Return testCases

Function CompareSolutions():
    testCases = GenerateTestCases()
    For each testCase in testCases:
        slowResult = MaxSubarraySum_N2(testCase, len(testCase))
        fastResult = MaxSubarraySum_Kadane(testCase, len(testCase))
        Assert(slowResult == fastResult)
    Print("All test cases passed!")
```

2.4 Trường hợp đặc biệt của test case

- Dãy số toàn số âm: Kiểm tra xem thuật toán có xử lý đúng không (trả về số âm lớn nhất).
- Dãy số chỉ có một phần tử: Xác minh kết quả là chính nó.
- Dãy số có giá trị rất lớn hoặc rất nhỏ (gần giới hạn của `int`).
- Dãy số với các giá trị luân phiên dương và âm để kiểm tra tính ổn định của thuật toán.