



VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

BÀI TẬP KIỂM TRA TÍNH ĐÚNG ĐẲN
ĐO HIỆU NĂNG CỦA CHƯƠNG TRÌNH

CS112.P11.KHTN

Sinh Viên :

Nguyễn Văn Minh - 23520945

Đồng Quốc Thắng - 23521421

Giảng viên :

Nguyễn Thanh Sơn

Ngày 4 tháng 11 năm 2024

Mục lục

1	Bài 1	2
1.1	Mã giả cho bài toán	2
1.2	Áp dụng các phương pháp kiểm thử	2
2	Bài 2	4
2.1	Giải thuật nặng	4
2.2	Giải thuật tối ưu	4

Chương 1

Bài 1

1.1 Mã giả cho bài toán

```
1 Function CalculateTotalPriceBeforeDiscount(Order order)
2     totalPrice = 0
3     For each product in order.Products:
4         totalPrice += product.Price * product.Quantity
5     Return totalPrice
6
7 Function CalculateTotalPriceWithDiscount(Order order)
8     totalPrice = 0
9     For each product in order.Products:
10        discountAmount = product.Price * product.Quantity * product.
            DiscountPercentage / 100
11        totalPrice += (product.Price * product.Quantity) - discountAmount
12    Return totalPrice
13
14 Function TinhChiPhi(Order order)
15     totalPriceWithDiscount = CalculateTotalPriceWithDiscount(order)
16     If order.IsRegularCustomer:
17         totalPriceWithDiscount *= 0.9
18     If CalculateTotalPriceBeforeDiscount(order) < 1000000:
19         totalPriceWithDiscount += order.ShippingFee
20     Return totalPriceWithDiscount
```

1.2 Áp dụng các phương pháp kiểm thử

Unit Test:

- **Phần cần kiểm thử:** Các hàm tính toán riêng lẻ (CalculateTotalPriceBeforeDiscount, CalculateTotalPriceWithDiscount, và logic áp dụng chiết khấu/vận chuyển trong TinhChiPhi).
- **Đặc điểm test cases:**
- **Input/Expected Output:**



- **CalculateTotalPriceBeforeDiscount:**
 - * **Input:** Danh sách sản phẩm (giá, số lượng).
 - * **Expected Output:** Tổng giá trị trước giảm giá.
- **CalculateTotalPriceWithDiscount:**
 - * **Input:** Danh sách sản phẩm (giá, số lượng, giảm giá).
 - * **Expected Output:** Tổng giá trị sau giảm giá.
- **TinhChiPhi** (chỉ logic chiết khấu và vận chuyển):
 - * **Input:** Trạng thái khách hàng, tổng giá trị đơn hàng trước/với giảm giá.
 - * **Expected Output:** Tổng chi phí cuối cùng.
- **Test Scenarios:**
 - Giá trị dương, âm, và 0 cho giá sản phẩm, số lượng, giảm giá.
 - Trạng thái khách hàng thường xuyên và không thường xuyên.
 - Tổng giá trị đơn hàng lớn hơn, nhỏ hơn, và bằng 1 triệu.

White box test

- **Phần cần kiểm thử:** Luồng quyết định và logic áp dụng giảm giá, chiết khấu, và phí vận chuyển trong **TinhChiPhi**.
- **Đặc điểm test cases:**
 - **Coverage mục tiêu:**
 - * **Decision Coverage:** Kiểm tra tất cả nhánh (if/else) liên quan đến giảm giá sản phẩm, chiết khấu khách hàng, và phí vận chuyển được thực thi.
 - * **Condition Coverage:** Mỗi điều kiện trong quyết định (ví dụ: **IsRegularCustomer**, tổng giá trị đơn hàng so với 1 triệu) được kiểm tra với cả giá trị **True** và **False**.
 - **Test Scenarios:**
 - * Đơn hàng với sản phẩm có giảm giá và không giảm giá.
 - * Khách hàng thường xuyên và không thường xuyên với tổng giá trị đơn hàng khác nhau (dưới, trên, bằng 1 triệu).

Chương 2

Bài 2

2.1 Giải thuật nặng

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int N = 2e5 + 10;
5 int n, sum, ans, A[N];
6
7 int main() {
8     ios::sync_with_stdio(0); cin.tie(0);
9     freopen("file.inp", "r", stdin);
10    freopen("file.ans", "w", stdout);
11
12    cin >> n;
13    for (int i = 1; i < n; i++) cin >> A[i];
14
15    ans = -2e9;
16    for (int i = 1; i <= n; i++) {
17        for (int j = i; j <= n; j++) {
18            sum = 0;
19            for (int z = i; z <= j; z++) {
20                sum += A[z];
21            }
22            ans = max(ans, sum);
23        }
24    }
25
26    cout << ans;
27 }
```

2.2 Giải thuật tối ưu

```
1 #include <bits/stdc++.h>
2 using namespace std;
```



```
3
4 const int N = 2e5 + 10;
5 int n, sum, ans, A[N], pre_sum[N];
6
7 int main() {
8     ios::sync_with_stdio(0); cin.tie(0);
9     freopen("file.inp", "r", stdin);
10    freopen("file.out", "w", stdout);
11
12    cin >> n;
13    for (int i = 1; i <= n; i++) {
14        cin >> A[i];
15        pre_sum[i] = A[i] + pre_sum[i - 1];
16    }
17
18    ans = -2e9;
19
20    // O(N)
21    int mi = 0;
22    for (int i = 1; i <= n; i++) {
23        ans = max(ans, pre_sum[i] - mi);
24        mi = min(mi, pre_sum[i]);
25    }
26
27    cout << ans;
28 }
```