

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ
MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
Ngày 26, Tháng 10, Năm 2024



LÀM BÀI TẬP NHÓM 3

Giáo viên hướng dẫn: *Nguyễn Thanh Sơn*

Nhóm 2: *Trần Vinh Khánh*, *Nguyễn Thiện Nhân*



Mục lục

1	Bài 1	2
1.1	Mô tả	2
1.2	Quy tắc	2
1.3	Mã giả	2
1.4	Kiểm thử	3
1.5	Các trường hợp đặc biệt:	3
2	Bài 2	3
2.1	Mô tả	3
2.2	Thuật toán với độ phức tạp $O(n^2)$	4
2.3	Thuật toán với độ phức tạp $O(n)$	4
2.4	Sinh bộ test và so sánh kết quả	5
2.5	Các Trường Hợp Kiểm Thử Đặc Biệt	5
2.6	Kết luận	5



1. Bài 1

1.1. Mô tả

Giả sử bạn đang phát triển một hệ thống quản lý đơn hàng cho một cửa hàng trực tuyến. Hệ thống cần tính toán tổng chi phí của một đơn hàng dựa trên các yếu tố sau:

- Thông tin đơn hàng:
 - + Danh sách sản phẩm: Mỗi sản phẩm có giá, số lượng và phần trăm giảm giá.
 - + Phí vận chuyển: Thêm phí nếu tổng giá trị đơn hàng nhỏ hơn 1 triệu.
 - + Loại khách hàng: Khách hàng thường xuyên sẽ được chiết khấu 10% trên tổng giá trị đơn hàng.
- Yêu cầu: Viết hàm `TinhChiPhi(Order order)` để tính tổng chi phí của đơn hàng sau khi đã áp dụng các quy tắc giảm giá, chiết khấu và vận chuyển.
- Order gồm:
 - + Danh sách các sản phẩm: Mỗi sản phẩm có giá gốc, số lượng và phần trăm giảm giá.
 - + `IsRegularCustomer`: Biến boolean xác định khách hàng thường xuyên hay không.
 - + `ShippingFee`: Phí vận chuyển.

1.2. Quy tắc

- Có 1 function tính tổng giá thành khi áp dụng giảm giá và 1 function không áp dụng giảm giá.
- Nếu giá trị đơn hàng trước khi áp dụng giảm giá lớn hơn hoặc bằng 1 triệu, miễn phí vận chuyển.
- Nếu khách hàng là khách hàng thường xuyên, chiết khấu 10% trên tổng giá trị đơn hàng sau khi áp dụng các giảm giá của sản phẩm.

1.3. Mã giả

```
1 function TinhTongKhongGiamGia(DonHang)
2     tong    0
3     for sanPham trong DonHang.danhSachSanPham do
4         tong    tong + (sanPham.gia  sanPham.soLuong)
5     end for
6     return tong
7 end function
8
9 function TinhTongCoGiamGia(DonHang)
10    tong    0
11    for sanPham trong DonHang.danhSachSanPham do
12        giaSauGiam    sanPham.gia  (1 - sanPham.phanTramGiamGia / 100)
13        tong    tong + (giaSauGiam  sanPham.soLuong)
14    end for
15    return tong
16 end function
17
```



```
18 function TinhChiPhiDonHang(DonHang)
19     tongTruocGiamGia    TinhTongKhongGiamGia(DonHang)
20     tongSauGiamGia      TinhTongCoGiamGia(DonHang)
21
22     if tongTruocGiamGia    1000000 then
23         chiPhiCuoiCung    tongSauGiamGia
24     else
25         chiPhiCuoiCung    tongSauGiamGia + DonHang.phiVanChuyen
26     end if
27
28     if DonHang.laKhachHangThuongXuyen then
29         chiPhiCuoiCung    chiPhiCuoiCung * 0.9
30     end if
31
32     return chiPhiCuoiCung
33 end function
```

1.4. Kiểm thử

- Unit Test: Kiểm thử từng hàm riêng biệt để đảm bảo tính chính xác của các phần:
 - + TinhTongKhongGiamGia: Kiểm tra tổng giá trị sản phẩm trước giảm giá.
 - + TinhTongCoGiamGia: Kiểm tra tổng giá trị sản phẩm sau giảm giá.
 - + TinhChiPhiDonHang: Kiểm tra tổng chi phí cuối cùng với các trường hợp miễn phí vận chuyển và chiết khấu cho khách hàng thường xuyên.

- White Box Test:

Kiểm thử mã nguồn bằng cách kiểm tra tất cả các nhánh điều kiện:

- + Trường hợp tổng giá trị đơn hàng trước giảm giá lớn hơn hoặc bằng 1 triệu để kiểm tra miễn phí vận chuyển.
- + Trường hợp khách hàng là khách hàng thường xuyên để kiểm tra chiết khấu 10%.

- Black Box Test

Kiểm thử dựa trên đầu vào và đầu ra mà không xem mã nguồn:

- + Đầu vào: Các bộ dữ liệu đơn hàng với giá trị, số lượng, và giảm giá khác nhau.
- + Đầu ra: Tổng chi phí đơn hàng tính đúng với các quy tắc về giảm giá và chiết khấu.

1.5. Các trường hợp đặc biệt:

- Tổng giá trị đơn hàng trước khi giảm giá đúng bằng 1 triệu để kiểm tra điều kiện miễn phí.
- Danh sách rỗng.
- Giảm giá 100%. vận chuyển.

2. Bài 2

2.1. Mô tả

Cho một dãy số nguyên a_1, a_2, \dots, a_n , yêu cầu tìm dãy con liên tiếp có tổng lớn nhất. Hai chỉ số (l, r) cần được xác định sao cho $1 \leq l \leq r \leq n$ và tổng $a_l + a_{l+1} + \dots + a_r$ là lớn nhất.

2.2. Thuật toán với độ phức tạp $O(n^2)$

Tính tổng của từng dãy con và cập nhật tổng lớn nhất nếu tìm thấy một tổng lớn hơn.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n;
7     long long ans = -1e18; // S dùng long long cho giá trị lớn
8     cin >> n;
9     vector<int> a(n + 5);
10
11     for (int i = 1; i <= n; i++) {
12         cin >> a[i];
13     }
14
15     for (int i = 1; i <= n; i++) {
16         int sum = 0;
17         for (int j = i; j <= n; j++) {
18             sum += a[j];
19             if (ans < sum) {
20                 ans = sum;
21             }
22         }
23     }
24
25     cout << ans;
26     return 0;
27 }
```

2.3. Thuật toán với độ phức tạp $O(n)$

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5
6 int main() {
7     int n;
8     long long ans = -1e18, pref = 0; // S dùng long long cho ans để đảm bảo độ
9     // lớn
10     cin >> n;
11     vector<int> a(n + 1); // ch s t 1 đến n (s dùng n+1 thay vì n+5)
12
13     for (int i = 1; i <= n; i++) {
14         cin >> a[i];
15     }
16
17     long long sum = 0;
18     for (int i = 1; i <= n; i++) {
19         sum += a[i];
20         ans = max(ans, sum - pref);
21         pref = min(pref, sum);
22     }
23 }
```



```
22  
23     cout << ans;  
24     return 0;  
25 }
```

2.4. Sinh bộ test và so sánh kết quả

```
1 Function TaoBoTest(numBoTest, giaTriNToiDa, giaTriTuyetDoiToiDa):  
2     danhSachBoTest = []  
3     For dem = 1 to numBoTest:  
4         kíchThuocMang = Random(1, giaTriNToiDa)  
5         mang = []  
6         For chiSo = 1 to kíchThuocMang:  
7             giaTri = Random(-giaTriTuyetDoiToiDa, giaTriTuyetDoiToiDa)  
8             mang.append(giaTri)  
9         danhSachBoTest.append(mang)  
10    Return danhSachBoTest
```

Mã giả sinh test

2.5. Các Trường Hợp Kiểm Thử Đặc Biệt

- Dãy chỉ chứa số âm.
- Dãy có cả số dương và số âm, kiểm tra tổng lớn nhất nằm ở giữa.
- Dãy có độ dài lớn nhất gần sát $n = 10^5$.

2.6. Kết luận

Thuật toán $O(n)$ là tối ưu.