

Giải thuật toán: Patrick and Shopping

Tư duy thuật toán theo triết lý "Learning How to Learn"

Coach Tư Duy Thuật Toán

Ngày 13 tháng 1 năm 2026

1. Phẫu thuật đề bài (Deconstruct)

Góc nhìn hình học

Quên cốt truyện đi, hãy nhìn bài toán dưới dạng hình học:

- **3 Điểm:** Nhà (Xuất phát/Đích), Shop 1, Shop 2.
- **Các cạnh (Quãng đường):**
 - d_1 : Nhà \leftrightarrow Shop 1
 - d_2 : Nhà \leftrightarrow Shop 2
 - d_3 : Shop 1 \leftrightarrow Shop 2

Nhiệm vụ

Xuất phát từ **Nhà** \rightarrow Ghé thăm **cả 2 shop** \rightarrow Quay về **Nhà**.

Mục tiêu: Tìm tổng quãng đường nhỏ nhất (min).

2. Vòng lặp tư duy - Chunk 1: Kịch bản di chuyển

Logic (Án dụ): Vẽ đường nét liền đi qua 3 điểm.

Có 3 kịch bản sơ khai:

① Đi vòng tròn:

Nhà → Shop 1 → Shop 2 → Nhà

② Đi kiểu "Con thoi"(Về nhà đổi hướng):

Nhà → Shop 1 → Nhà . . . rồi . . . Nhà → Shop 2 → Nhà

③ Đi kiểu "Kẹp nách"(Đi ké):

Nhà → Shop 1 → Shop 2 → Shop 1 → Nhà

Bẫy (Trap)

Đừng mặc định đi vòng tròn ($d_1 + d_2 + d_3$) là nhanh nhất. Nếu d_3 quá lớn, đi về nhà rồi sang shop kia còn lợi hơn.

Thử thách tư duy 1 (Mental Check)

Dữ liệu: $d_1 = 10, d_2 = 20, d_3 = 100$ (Đường nối 2 shop siêu xa).

Tính toán các phương án:

- **Cách A (Vòng tròn):** $d_1 + d_3 + d_2 = 10 + 100 + 20 = 130$
- **Cách B (Con thoi):** $2(d_1) + 2(d_2) = 20 + 40 = \mathbf{60}$
- **Cách C (Đi ké):** $2(d_1) + 2(d_3) = 20 + 200 = 220$

Kết luận

Chọn Cách B (60).

Chứng minh: Đôi khi đường thẳng nối 2 điểm (d_3) là một cái "bẫy" chi phí cao.

3. Vòng lặp tư duy - Chunk 2: Bức tranh toàn cảnh

Chúng ta đã bỏ sót một khả năng ở Chunk 1.

Khi đi kiểu "Kẹp nách", tại sao lại chỉ ưu ái Shop 1?

- Thay vì: Nhà → Shop 1 → Shop 2 ...
- Ta có thể: Nhà → Shop 2 → Shop 1 ...

Tổng kết 4 Chiến thuật di chuyển:

- ➊ **Vòng tròn:** $Cost = d_1 + d_2 + d_3$
- ➋ **Con thoi (Đi riêng):** $Cost = 2(d_1 + d_2)$
- ➌ **Ké qua Shop 1:** $Cost = 2(d_1 + d_3)$
- ➍ **Ké qua Shop 2:** $Cost = 2(d_2 + d_3)$

Thử thách tư duy 2 (Mental Check)

Dữ liệu: $d_1 = 10$, $d_2 = 100$ (Xa tít), $d_3 = 5$ (Hai shop cạnh nhau).

Phân tích:

- Nếu đi theo thói quen (Vòng tròn): $10 + 100 + 5 = 115$.
- Nếu dùng chiến thuật **Ké qua Shop 1**:

$$2 \times (d_1 + d_3) = 2 \times (10 + 5) = \mathbf{30}$$

Kết luận

Đáp án là **30**.

Con đường trực tiếp d_2 (100) giống như trạm thu phí giá cắt cổ. Đi vòng qua d_1 và d_3 rẻ hơn nhiều.

4. Chốt hạ thuật toán (The Solution)

Không có một công thức duy nhất đúng cho mọi trường hợp.

Giải pháp: Brute-force (Vét cạn). Tính tất cả 4 trường hợp và chọn Min.

Công thức tổng quát

Gọi ans là kết quả cần tìm:

$$ans = \min \begin{cases} d_1 + d_2 + d_3 & (\text{Vòng tròn}) \\ 2(d_1 + d_2) & (\text{Con thoi}) \\ 2(d_1 + d_3) & (\text{Qua Shop 1}) \\ 2(d_2 + d_3) & (\text{Qua Shop 2}) \end{cases}$$

Thử thách cuối cùng (Mã giả):

- Bước 1: Nhập d_1, d_2, d_3 .
- Bước 2: Tính 4 biến a, b, c, d theo công thức trên.
- Bước 3: Output → Dùng hàm **min(a, b, c, d)**.

Hiện thực hóa: Python Code

Python hỗ trợ hàm `min()` với nhiều tham số rất tiện lợi.

```
1 # B      c  1: N h p      d      l i u
2 d1, d2, d3 = map(int, input().split())
3
4 # B      c  2 & 3: T nh      to n v      in ra min ngay      l p      t c
5 ans = min(
6     d1 + d2 + d3,          # Cach 1: Vong tron
7     2 * (d1 + d2),        # Cach 2: Di le tung cai
8     2 * (d1 + d3),        # Cach 3: Ke qua shop 1
9     2 * (d2 + d3)         # Cach 4: Ke qua shop 2
10)
11
12 print(ans)
```

Hiện thực hóa: C++ Code

Với C++11 trở lên, dùng `min(...)` để so sánh danh sách.

```
1 #include <iostream>
2 #include <algorithm> // Thu vien chua ham min
3 using namespace std;
4
5 int main() {
6     long long d1, d2, d3;
7     // Dung long long phong khi tong > 2 ty
8     cin >> d1 >> d2 >> d3;
9
10    long long opt1 = d1 + d2 + d3;
11    long long opt2 = 2 * (d1 + d2);
12    long long opt3 = 2 * (d1 + d3);
13    long long opt4 = 2 * (d2 + d3);
14
15    // Tim min trong danh sach khoi tao
16    cout << min({opt1, opt2, opt3, opt4});
17
18    return 0;
19 }
```

Bài học rút ra (Key Takeaway)

① Đừng tin vào trực giác hình học trên giấy:

- Toán học: $a + b > c$ (Bất đẳng thức tam giác).
- Thực tế (Giao thông/Đồ thị): Đường thẳng nối 2 điểm (d_3) có thể xa vô lý (tắc đường, đường xá).

② Sức mạnh của Vét cạn (Brute-force):

- Khi số lượng trường hợp ít (chỉ 4 cách), đừng cố tìm công thức toán học phức tạp để loại trừ.
- Hãy để máy tính tính hết và chọn cái tốt nhất.

Chúc mừng bạn đã chinh phục bài 599A!