

## Bài tập về nhà - Nhóm 3

Thành viên:

1. Hà Xuân Thiện - 24520031
2. Trần Quang Trường - 24521901

Ngày 5 tháng 12 năm 2025

Ta có một số nhận xét bài toán như sau:

Khi một Pokemon  $u$  đánh thắng  $v$  theo thuộc tính  $x$  thì:

- Hoặc  $a_{u,x} \geq a_{v,x}$ , lúc này chi phí bỏ ra  $k_x(u, v) = 0$ .
- Hoặc ta cần tốn chi phí  $k_x(u, v) = a_{v,x} - a_{u,x}$  để  $a_{u,x} + k_x(u, v) \geq a_{v,x}$ .

Mở rộng hơn, để một Pokemon  $u$  đánh thắng  $v$  thì ta cần tốn chi phí là  $\min_{x=1}^m k_x(u, v)$ .

Lúc này, biến đổi bài toán thành bài toán đồ thị (**phương pháp thiết kế biến đổi để trị**). Cụ thể:

- coi mỗi pokemon là một đỉnh của đồ thị.
- Với mỗi cạnh  $u$  đi tới  $v$  có trọng số bằng  $\min_{x=1}^m k_x(u, v) + c_v$ .
- Bài toán trở thành tìm kiếm đường đi ngắn nhất từ 1 tới  $n$ .

Để giải quyết bài toán tìm kiếm đường đi ngắn nhất từ 1 tới  $n$  ta sử dụng thuật toán dijkstra (**phương pháp thiết kế tham lam**).

Tuy nhiên, khi đánh giá về độ phức tạp thời gian của bài toán hiện tại, để xây dựng được tập cạnh, tại mỗi đỉnh  $u$  ta cần xét trọng số tại mỗi đỉnh  $v$  với chi phí  $O(m)$ , do đó tổng độ phức tạp thời gian để xây dựng tập cạnh là  $O(n^2 \cdot m)$ .

Người thông minh như bạn sẽ nghĩ tới việc tối ưu tập cạnh của bài toán bằng cách thêm một tập đỉnh chứa thuộc tính của mỗi pokemon nên tập đỉnh lúc này là  $n \cdot m + n$ :

- $B_i$  là đường đi ngắn nhất từ 1 đến  $i$ .
- $A_{i,x} = \min_{u=1}^n [B_u + k_x(u, i)]$ .

Tập cạnh lúc này trở thành:

1. Tại mỗi thuộc tính  $x$ , không mất tính tổng quát, ta giả sử  $a_{1,x}, a_{2,x}, \dots, a_{n,x}$  tăng dần. (aka Phương pháp thiết kế biến đổi để trị)
  - Với mỗi  $i \geq 2$ , ta thêm 1 cạnh từ  $A_{i,x}$  đến  $A_{i-1,x}$  với trọng số bằng 0.

- Với mỗi  $i < n$ , ta thêm 1 cạnh từ  $A_{i,x}$  đến  $A_{i+1,x}$  với trọng số  $a_{i+1,x} - a_{i,x}$ .
- 2. Tại mỗi  $1 \leq i \leq n$ , với mỗi  $1 \leq x \leq m$ , ta thêm cạnh  $A_{i,x}$  đến  $B_i$  với trọng số là  $c_i$ ; và một cạnh từ  $B_i$  đến  $A_{i,x}$  với trọng số là 0.

Như vậy đồ thị trở thành  $G = (V = n \cdot m + n, E = 4 \cdot n \cdot m)$ . Bài toán trở thành tìm đường đi ngắn nhất từ  $B_1$  đến  $B_n$ .

Vì ta sử dụng thuật toán dijkstra do đó:

- **Độ phức tạp thời gian** là  $\Theta(n \cdot m \log(n \cdot m))$ .

Ta cần lưu trữ tập đỉnh  $n \cdot m + n$  cho thuật toán nên:

- **Độ phức tạp không gian của bài toán** là  $\Theta(n \cdot m)$ .

**Vậy, tại sao cách tối ưu tập đồ thị như trên lại đúng?**

Khi tập cạnh là mỗi pokemon có thuộc tính thì việc tạo mỗi cạnh  $A_{u,x}$  đến mọi  $A_{v,x}$  với  $1 \leq v \leq n$  là không cần thiết.

Nếu xây dựng theo như 1, thì  $A_{u,x}$  có thể đến  $A_{u+2,x}$  với trọng số là

$$a_{u+2,x} - a_{u+1,x} + a_{u+1,x} - a_{u,x} = a_{u+2,x} - a_{u,x}$$

bằng cách đi qua đỉnh trung gian  $A_{u+1,x}$ .

Mở rộng,  $A_{u,x}$  có thể đến  $A_{v,x}$  với trọng số  $A_{v,x} - A_{u,x}$  thông qua các đỉnh trung gian  $A_{u+1,x}, A_{u+2,x}, \dots, A_{v-1,x}$  với  $u < v$ .

Cũng tương tự, ta có thể chứng minh rằng:

$A_{u,x}$  có thể đến  $A_{v,x}$  với trọng số bằng 0 thông qua các đỉnh trung gian  $A_{u-1,x}, A_{u-2,x}, \dots, A_{2,x}$ .

**Vì sao phương pháp thiết kế lại phù hợp?**

- **Với phương pháp thiết kế biến đổi để trị**, như đã trình bày qua cách giải quyết bài toán, vì bài toán gốc sẽ khó giải quyết nếu ta không biến đổi bài toán và việc biến đổi bài toán ban đầu thành bài toán khác không làm khác đi kết quả bài toán. Như việc biến đổi bài toán thành bài toán đồ thị không làm mất đi tính chất của bài toán; việc sắp xếp lại mảng tăng dần cũng không làm thay đổi bài toán.
- **Với phương pháp thiết kế tham lam**, vì đơn giản thuật toán ta sử dụng là thuật toán dijkstra, phần này chúng ta đã có bàn ở buổi lý thuyết đồ thị nên mình sẽ không giải thích thêm.