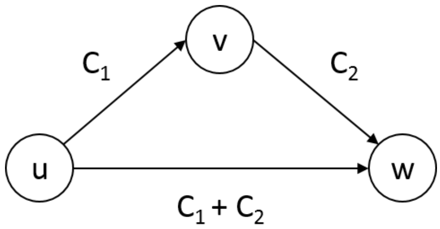
SEQOP

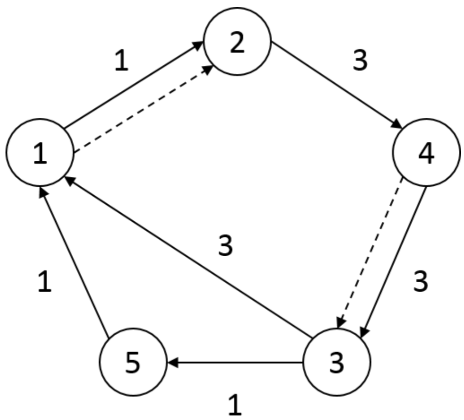
Nguồn: Kyoto University Programming Contest 2014 - Problem I

Trước hết, với mỗi loại phép biến đổi (X, Y, C), thay vì tăng A_X thêm 2, giữ nguyên A_Y và tăng các giá trị A còn lại thêm 1, ta sẽ tăng A_X thêm 1, giảm A_Y đi 1 và giữ nguyên các giá trị A còn lại. Khi đó, điều kiện "các phần tử trong dãy A bằng nhau" trở thành "các phần tử trong dãy A đều có giá trị 0".

Nhận xét, việc thực hiện hai phép biến đổi (u, v, C_1) và (v, w, C_2) tương đương với việc thực hiện một phép biến đổi $(u, w, C_1 + C_2)$. Nếu ta tưởng tượng mỗi phần tử trong dãy là một đỉnh, mỗi phép biến đổi là một cạnh thì việc thực hiện phép biến đổi $(u, w, C_1 + C_2)$ tương đương với đường đi $u \to v \to w$ với tổng trọng số $C_1 + C_2$.



Với M loại phép biến đổi đã cho, ta có thể xây dựng một đồ thị gồm N đỉnh, M cạnh, cạnh thứ i nối từ X_i đến Y_i với trọng số C_i. Hình vẽ sau mô tả đồ thị tương ứng với test ví dụ đầu tiên (M cạnh nét liền tương ứng M loại phép biến đổi, K cạnh nét đứt tương ứng với K phép biến đổi đã được thực hiện):



Khi đó, bài toán trở thành: tìm một tập các chu trình (không nhất thiết là chu trình đơn) bao phủ K cạnh $P_1, P_2, ..., P_K$ sao cho tổng trọng số của các chu trình là nhỏ nhất có thể (lưu ý, nếu một cạnh nào đó xuất hiện x lần trong dãy P thì cạnh đó cũng phải xuất hiện ít nhất x lần trong tập các chu trình).

Với subtask 1 ($K = 1$), đáp án chính là trọng số của đường đi ngắn nhất (có tổng trọng số nhỏ nhất) từ Y_{P_1} đến X_{P_1}.

Với subtask 2, trước hết, ta tính $dist[i][u]$ là đường đi ngắn nhất từ Y_{P_i} đến đỉnh u bằng thuật toán Dijkstra.

Đến đây, ta có hai thể giải bài này bằng hai cách:

Cách 1 - Quy hoạch động trạng thái

Trước hết, ta tính $dist[i][u]$ là đường đi ngắn nhất từ Y_{P_i} đến đỉnh u bằng thuật toán Dijkstra.

Tiếp theo, ta tính $cyc[mask]$ là chu trình có tổng trọng số nhỏ nhất bao phủ tập cạnh S được biểu diễn bởi $mask$ (nếu bit thứ i của $mask$ bằng 1 thì cạnh P_i thuộc S, ngược lại thì cạnh P_i không thuộc S).

Để tính mảng cyc, ta sẽ duyệt qua từng cạnh s trong mảng P và dùng thuật toán quy hoạch động sau:

Gọi $dp[mask][i]$ là đường đi có tổng trọng số nhỏ nhất bao phủ tập cạnh được biểu diễn bởi $mask$ và kết thúc tại cạnh P_i. Ta khởi tạo $dp[2^s][s] = 0$ và cập nhật trạng thái quy hoạch động theo công thức:

$$dp[mask][i] = dp[mask \oplus 2^j][j] + dist[j][P_{X_i}] \text{ với mọi } j \text{ sao cho } mask \text{ không chứa } j$$

Khi đã có mảng cyc, ta tiếp tục dùng quy hoạch động để tìm đáp án. Gọi $f[mask]$ là tổng trọng số nhỏ nhất của một tập chu trình bao phủ tập cạnh được biểu diễn bởi $mask$. Ta có công thức quy hoạch động sau:

$$f[mask] = min(cyc[mask], f[sub] + f[mask \oplus sub]) \text{ với mọi } sub \text{ là tập con của } mask$$

Độ phức tạp: $O((N + M)\log N + 3^K)$

Cách 2 - Luồng cực đại chi phí cực tiểu

Ta xây dựng một mạng gồm $N + 2$ đỉnh, bao gồm:

- N đỉnh đại diện cho các phần tử trong dãy số
- Đỉnh phát s
- Đỉnh thu t

Ta thêm cạnh vào mạng trên theo quy tắc sau:

- Với mỗi phép biến đổi P_i , ta thêm hai cạnh sau:
  - Cạnh từ s đến Y_{P_i} với khả năng thông qua 1 và chi phí cho từng đơn vị luồng 0.
  - Cạnh từ X_{P_i} đến t với khả năng thông qua 1 và chi phí cho từng đơn vị luồng 0.
- Với mỗi cặp chỉ số (i, j) với $1 \leq i, j \leq K$, ta thêm cạnh từ Y_{P_i} đến X_{P_j} với khả năng thông qua vô cực (ta có thể đặt là 10^9) và chi phí cho từng đơn vị luồng $dist[i][X_{P_j}]$

Khi đó, đáp án cần tìm chính là luồng cực đại chi phí cực tiểu của mạng trên.

Có thể tham khảo cách cài đặt thuật toán tìm luồng cực đại chi phí cực tiểu với độ phức tạp $O(min(E^2 * V^2, E * V * F))$ (với V là số đỉnh, E là số cạnh, F là giá trị luồng cực đại) tại `https://sites.google.com/site/indy256/algo/min_cost_flow_bf`

Độ phức tạp: $O((N + M) \log N + K^4)$

Tag: Shortest Path, Dynamic Programming, Bitmask, Flow