

Trại Đông Đà Lạt

Hướng dẫn giải bài tập

29-31/10/2023

Ngày 29 tháng 10 năm 2023

Bài 1. Mã hoá từ – CIPHER

Bài 2. Mô hình trái cây – PARADIGM

Bài 3. Giá trị đối xứng – PALIVAL

Cipher

Công chỉ chuyển mỗi lần một từ cho Sao và từ đó chỉ bao gồm các chữ cái viết hoa. Mỗi từ được mã hóa như sau:

- ▶ Công gán một số cho mỗi chữ cái dựa trên vị trí của chữ cái đó trong bảng chữ cái, trong đó $A = 0, B = 1, \dots, Z = 25$.
- ▶ Đối với mỗi chữ cái trong từ, Công xác định giá trị được mã hóa của chữ cái đó bằng cách tính tổng giá trị của 1 hoặc 2 chữ cái liền kề với chữ cái đó trong từ. Công lấy tổng đó theo modulo 26, và gán thành giá trị mới của chữ cái. Sau đó, Công chuyển đổi giá trị này trở lại thành chữ in hoa dựa trên các vị trí trong bảng chữ cái, như trước đây.
- ▶ Từ được mã hóa được xác định bằng cách mã hóa từng chữ cái trong từ bằng phương pháp này. Mã hóa mỗi chữ cái chỉ dựa trên các chữ cái trong tin nhắn ban đầu không được mã hóa chứ không dựa trên bất kỳ chữ cái nào đã được mã hóa.

Yêu cầu: Hãy giúp Sao giải mã tin nhắn của Công.

Subtask 1: $2 \leq \text{độ dài của } W \leq 4$

- ▶ Chỉ có $26^2 + 26^3 + 26^4 = 475228$ từ có độ dài trong phạm vi $[2, 4]$. Vì vậy ta có thể mã hóa tất cả các từ có thể và sau đó tạo một từ điển ngược. Ví dụ, khi thấy SOUP mã hóa thành OMDU, ta có thể tạo một mục trong từ điển ngược của mình với khóa OMDU và giá trị SOUP. Nếu thấy rằng một từ đã được mã hóa thành một khóa mà đã có trong từ điển (ví dụ: ABC và CBA đều mã hóa thành BCB), thì việc giải mã khóa đó là nhập nhằng và có thể thay thế giá trị bằng AMBIGUOUS.
- ▶ Sau khi hoàn thành từ điển ngược, ta chỉ tra cứu từng từ trong từ điển và ghi ra giá trị cho câu trả lời tương ứng.

Subtask 2: $2 \leq \text{độ dài của } W \leq 50$. Trường hợp 1: Các từ có độ dài chẵn

- ▶ Khi một từ được mã hóa, mọi chữ cái trong từ đó (trừ chữ cái đầu tiên và chữ cái cuối cùng) đều được mã hóa bằng cách tính tổng hai chữ cái liền kề với chữ cái đó.
- ▶ Đối với các chữ cái đầu tiên và cuối cùng, giá trị được mã hóa chỉ là giá trị không được mã hóa của một chữ cái liền kề với nó.
- ▶ Điều này có nghĩa là chúng ta có thể dễ dàng tìm thấy các chữ cái thứ hai và chữ cái sát với chữ cái cuối cùng của từ không mã hóa: chúng lần lượt là chữ cái đầu tiên và cuối cùng của từ mã hóa.
- ▶ Khi đã xác định những vị trí này trong từ được giải mã, ta có thể giải mã tiếp phần còn lại của từ được mã hóa!

Subtask 2: $2 \leq \text{độ dài của } W \leq 50$. Trường hợp 1: Các từ có độ dài chẵn

Ví dụ: giải mã từ được mã hóa ADMZZO như sau. Ta biểu thị một chữ cái E là từ được mã hoá và D là từ được giải mã, theo sau là một số cho biết vị trí của chữ cái trong từ, bắt đầu từ 1.

- ▶ Giá trị chữ cái của từ được mã hóa là: 0 3 12 25 25 14.
- ▶ Như đã giải thích ở trên, $D2$ và $D5$ tương ứng với $E1$ và $E6$, vì vậy từ được giải mã có các giá trị chữ cái dạng _ 0 _ _ 14 _.
- ▶ Bây giờ ta sẽ xác định các chữ cái ở vị trí số chẵn của từ được giải mã, đi từ trái sang phải. Bắt đầu bằng việc xác định $D4$. Ta đã biết rằng $D2$ có giá trị 0. Vì ta biết rằng $(D2 + D4) \% 26 = E3$, nên $D4 = (E3 - D2) \% 26$. $E3$ có giá trị 12, nên $D4 = (12 - 0) \% 26 = 12$. Ta có các chữ cái được giải mã là: _ 0 _ 12 14 _.
- ▶ Sau khi đã biết $D4$, ta có thể tìm $D6$ theo cách tương tự: $D6 = (E5 - D4) \% 26 = (25 - 12) \% 26 = 13$. Vậy ta có _ 0 _ 12 14 13.

Subtask 2: $2 \leq \text{độ dài của } W \leq 50$. Trường hợp 1: Các từ có độ dài chẵn

- ▶ Sau đó, ta thực hiện tương tự từ phải sang trái, bắt đầu từ $D5$. Ta có $D3 = (E4 - D5)\%26 = (25 - 14) = 11$, và $D1 = (E2 - D3)\%26 = (3 - 11)\%26 = 18$. (Có thể lấy mô đun của một số âm, hoặc xử lý an toàn bằng cách luôn cộng thêm 26 nếu chỉ muốn xử lý các giá trị dương, ví dụ: $(3 - 11 + 26)\%26$). Ta có tất cả các giá trị chữ cái trong từ được giải mã: 18 0 11 12 14 13.
- ▶ Bây giờ ta chỉ cần chuyển đổi danh sách các giá trị chữ cái được giải mã này thành chữ in hoa, kết quả được từ giải mã SALMON.

Phương pháp này cho phép giải mã bất kỳ từ nào có độ dài chẵn. Phương pháp này là tất định, nghĩa là luôn giải mã được theo một cách duy nhất. Vì vậy bất kỳ từ được mã hóa nào có độ dài chẵn đều có chính xác một cách giải mã khả thi.

Subtask 2: $2 \leq \text{độ dài của } W \leq 50$. Trường hợp 2: Các từ có độ dài lẻ.

- ▶ Nhìn lại ví dụ BCB. Biết rằng $D2 = E1$ và $D2 = E3$, vậy $D2$ là B. Nhưng còn $D1$ và $D3$ thì sao? Ta biết là $(D1 + D3) \% 26 = E2$ với 26 cặp khác nhau, vì vậy chưa đủ thông tin để giải mã.
- ▶ Trên thực tế, đây là trường hợp của bất kỳ từ nào có độ dài lẻ có thể xuất hiện trong tập dữ liệu. Việc biết các chữ cái thứ hai và chữ cái gần cuối cùng của từ được giải mã không giúp ta giải mã tiếp được toàn bộ từ đó, vì cả hai chữ cái đó đều ở vị trí đánh số chẵn của từ được giải mã. Ta chưa có thông tin cho việc giải mã các vị trí số lẻ. Chữ cái đầu tiên của từ được giải mã có thể là bất kỳ chữ cái nào và mỗi lựa chọn sẽ xác định ra tất cả các chữ cái còn lại. Ta không có cách nào biết được khả năng nào trong số 26 khả năng mà Công đã mã hóa ban đầu.
- ▶ Đề bài cho biết các tập dữ liệu không bao gồm các từ như APE không có cách giải mã hợp lệ, do đó, bất kỳ từ nào có độ dài lẻ trong tập dữ liệu đều có nhiều cách giải mã. Do đó: đối với các từ có độ dài chẵn, sử dụng phương pháp trên và đối với các từ có độ dài lẻ, kết quả là AMBIGUOUS.

Bài 1. Mã hoá từ – CIPHER

Bài 2. Mô hình trái cây – PARADIGM

Bài 3. Giá trị đối xứng – PALIVAL

Mô hình trái cây – Paradigm

- ▶ Trên bàn tổng cộng N trái cây, trái thứ i có trọng lượng là w_i . Tuấn sử dụng $N - 1$ cái que, trong đó mỗi que sẽ cắm lên 2 trái cây khác nhau, tạo thành một đồ thị dạng cây, .
- ▶ Sau khi hoàn thành mô hình thì mẹ Tuấn đã bắt cậu phải gỡ ra. Khi Tuấn rút cái que cắm giữa 2 trái cây (u, v) thì đồ thị xem như xóa 1 cạnh nối giữa 2 đỉnh (u, v) và tạo ra 2 thành phần liên thông. Năng lượng tiêu thụ để rút cái que cắm giữa 2 trái cây (u, v) được tính bằng trọng lượng của trái cây lớn nhất nằm trong thành phần liên thông chứa u cộng với trọng lượng của trái cây lớn nhất nằm trong thành phần liên thông chứa v .
- ▶ **Yêu cầu:** Hãy giúp Tuấn tính giá trị tổng năng lượng tối thiểu phải sử dụng để rút tất cả các que ra khỏi mô hình.

Subtask 1 ($1 \leq N \leq 10$): $O(N! * \log N)$

Ta thực hiện duyệt hoán vị thứ tự của tất cả các cạnh sẽ xóa. Dựa vào thứ tự đó, thay vì lần lượt xóa từng cạnh thì ta thực hiện ngược lại bằng cách thêm từng cạnh và kết nối 2 thành phần liên thông. Sử dụng thuật toán DSU để làm điều trên.

Subtask 2 (Luôn có một que cắm giữa trái thứ i và trái thứ $i+1$): $O(N * \log N)$

- ▶ Nhận xét quan trọng: Ưu tiên xóa những cạnh nối với trái cây có trọng lượng lớn nhất. Như vậy ta có được thứ tự xử lý trên các trái cây $pos_1, pos_2, \dots, pos_N$ thỏa mãn $w_{pos_1} \geq w_{pos_2} \geq \dots \geq w_{pos_N}$.
- ▶ Sử dụng một tập hợp set chứa các đoạn (l, r) sau khi thực hiện xóa từng cạnh, mỗi đoạn (l, r) đại diện cho một thành phần liên thông kết nối từ trái cây thứ l đến trái cây thứ r .
- ▶ Khi xét trái cây thứ pos_i ta sẽ tìm đoạn (l, r) có chứa trái cây đó ($l \leq pos_i \leq r$), vì đồ thị trong trường hợp này là 1 đường thẳng nên ta xóa không quá 2 cạnh, lúc đó ta xóa đoạn (l, r) khỏi set và thêm vào set tối đa 2 đoạn mới $(l, pos_i - 1)$ và $(pos_i + 1, r)$. Kết quả cộng thêm w_{pos_i} nhân với số cạnh bị xóa.

Subtask 3 ($N \leq 1000$): $O(N^2)$

Ta dựa theo ý tưởng subtask 2 là xử lý từ các trái cây có trọng lượng lớn nhất về nhỏ nhất. Mỗi lần thực hiện xóa các cạnh thì cần đánh dấu các cạnh đã bị xóa và quản lý các thành phần liên thông được tách ra.

Subtask 4 ($N \leq 10^5$): $O(N * \log N)$

Thứ tự xóa cạnh ở subtask 2&3 là thực hiện xóa lần lượt các cạnh nối với các trái cây có trọng lượng lớn nhất về nhỏ nhất. Điều này có nghĩa, nếu ta làm ngược lại thứ tự trên, đó là thêm cạnh vào các cặp (u, v) có $\max(w_u, w_v)$ từ nhỏ nhất đến lớn nhất thì có thể sử dụng DSU để giải quyết như subtask 1. Chỉ cần sắp xếp các cạnh (u, v) theo thứ tự $\max(w_u, w_v)$ không giảm thì ta có được thứ tự thêm cạnh thỏa mãn.

Bài 1. Mã hoá từ – CIPHER

Bài 2. Mô hình trái cây – PARADIGM

Bài 3. Giá trị đối xứng – PALIVAL

Giá trị đối xứng – Palival

- ▶ Một chuỗi S gồm các ký tự in thường và cho phép thay đổi tối đa 1 ký tự trong chuỗi đó thành 1 ký tự khác. Yêu cầu của bài toán là hãy tính giá trị đối xứng lớn nhất có thể của chuỗi S sau khi thực hiện thay đổi tối đa 1 ký tự.
- ▶ Giá trị đối xứng của một chuỗi S bất kỳ là số cặp chỉ số (i, j) sao cho chuỗi con $S_i S_{i+1} \dots S_j$ là một chuỗi đối xứng.

Subtask 1 ($1 \leq |S| \leq 100$): $O(N^3 * 26)$)

Ta duyệt từng vị trí và thay thế kí tự ở vị trí đó thành kí tự khác. Sau đó duyệt các chỉ số (l, r) và kiểm tra xem chuỗi con $S_l S_{l+1} \dots S_r$ có phải là một chuỗi đối xứng hay không. Lấy kết quả là giá trị đối xứng lớn nhất của những lần duyệt vị trí và thay thế kí tự.

Subtask 2 ($1 \leq |S| \leq 5000$): $O(N^2)$)

Để đơn giản, ta chỉ xử lí trên các xâu đối xứng có độ dài lẻ, còn các xâu có độ dài chẵn thì làm tương tự. Đối với mỗi chỉ số a , ta xem nó là tâm của một xâu đối xứng độ dài lẻ, gọi r_a là giá trị lớn nhất thỏa mãn xâu $S_{a-r_a}S_{a-r_a+1}\dots S_{a+r_a}$ là một xâu đối xứng. Thực hiện tìm r_a bằng chập nhị phân với hash.

Tiếp theo, ta phải xác định được số lượng xâu đối xứng bị mất đi và số lượng xâu đối xứng xuất hiện thêm mỗi khi đổi kí tự thứ i sang 1 kí tự c khác. Đầu tiên xác định số lượng xâu đối xứng bị mất đi:

- ▶ Gọi $q[i]$ là số lượng xâu đối xứng bị mất đi khi thay đổi kí tự thứ i .
- ▶ Duyệt chỉ số a là tâm của xâu đối xứng độ dài lẻ, nếu kí tự thứ i với $i \in [a - r_a, a - 1]$ thay đổi thì sẽ có $i - (a - r_a) + 1$ xâu đối xứng bị mất đi. Khi đó ta tăng giá trị mảng $q[i] += i - (a - r_a) + 1$.
- ▶ Tương tự như trên, ta xác định được tổng số lượng xâu đối xứng bị mất đi khi thay đổi kí tự thứ i với $i \in [a + 1, a + r_a]$ là $(a + r_a) - i + 1$. Lúc này tăng giá trị mảng $q[i] += (a + r_a) - i + 1$.

Subtask 2 ($1 \leq |S| \leq 5000$): $O(N^2)$

Tiếp theo, xác định số lượng xâu đối xứng xuất hiện thêm:

- ▶ Sử dụng mảng $f[i][c]$ là số lượng xâu đối xứng xuất hiện thêm khi thay đổi kí tự thứ i thành kí tự c .
- ▶ Duyệt chỉ số a là tâm của xâu đối xứng độ dài lẻ, ta khẳng định được số lượng xâu đối xứng xuất hiện thêm nếu thay đổi kí tự thứ $i = a - r_a - 1$ thành kí tự thứ $i = a + r_a + 1$ hoặc ngược lại. Gọi r'_a là giá trị lớn nhất nếu như thay đổi kí tự thứ $i = a - r_a - 1$ thành kí tự thứ $i = a + r_a + 1$ hoặc ngược lại.
- ▶ Để tính r'_a chặt nhị phân từ giá trị $r_a + 2$ và sử dụng hash để kiểm tra xâu con $S_{a-r'_a} \dots S_{a-r_a-2}$ có bằng xâu con $S_{a+r_a+2} \dots S_{a+r'_a}$. Như vậy số lượng xâu đối xứng xuất hiện thêm khi thay đổi kí tự thứ i ($i = a - r_a + 1$ hoặc $i = a + r_a + 1$) thành kí tự c là $r'_a - r_a$. Cập nhật mảng $f[i][c] += r'_a - r_a$.

Với độ dài chẵn ta làm tương tự để tính các giá trị $q[i], f[i][c]$.

Gọi $base$ là giá trị đối xứng của xâu ban đầu. Thực hiện lấy giá trị đối xứng lớn nhất khi thay đổi kí tự thứ i thành kí tự c là:

- ▶ $ans = \max(ans, base + f[i][c] - q[i])$.

Subtask 3 ($N \leq 10^5$): $O(N * \log N)$

Ở subtask 2, phần tính giá trị $q[i]$ khiến ta phải mất độ phức tạp $O(N)$ nên ở subtask 3 ta sẽ tối ưu điểm này bằng cách sử dụng 2 mảng $k[i]$, $l[i]$ là 2 mảng cộng dồn. Theo subtask 2:

- ▶ Khi kí tự thứ i với $i \in [a - r_a, a - 1]$ thay đổi thì sẽ có $i - (a - r_a) + 1$ xâu đối xứng bị mất đi. Khi đó ta tăng giá trị mảng $k[i] + = 1$, $l[i] - = (a - r_a) + 1$ với $i \in [a - r_a, a - 1]$ bằng cách sau:
 - ▶ $k[a - r_a] + = 1$ và $k[a] - = 1$;
 - ▶ $l[a - r_a] - = (a - r_a) + 1$ và $l[a] + = (a - r_a) + 1$.
- ▶ Khi kí tự thứ i với $i \in [a + 1, a + r_a]$ thay đổi thì sẽ có $(a + r_a) - i + 1$ xâu đối xứng bị mất đi. Khi đó ta tăng giá trị mảng $k[i] - = 1$, $l[i] + = (a + r_a) + 1$ với $i \in [a + 1, a + r_a]$ bằng cách sau:
 - ▶ $k[a + 1] - = 1$ và $k[a + r_a + 1] + = 1$;
 - ▶ $l[a + 1] + = (a + r_a) + 1$ và $l[a + r_a + 1] - = (a + r_a) + 1$.
- ▶ Các giá trị $k[i]$, $l[i]$ ta sử dụng cộng dồn và lúc này $q[i] = k[i] * i + l[i]$.