Trại Đông Đà Lạt

Hướng dẫn giải bài tập

29-31/10/2023

Ngày 30 tháng 10 năm 2023

Bài 1. Giải mã – DECRYPT

Bài 2. Công thức nấu ăn – RECEIPT

Bài 3. Tom và Jerry – TOMJERRY

Giải mã – Decrypt

Định nghĩa 1 hàm $f(S) = K$ là một hàm mã hóa xâu kí tự S thành một số K. Hàm $f(S)$ được xây dựng bởi công thức:

- $f(S) = 0$ nếu xâu S rỗng.
- $f(S + ch) = ((f(S) * 33)\ XOR\ odr(ch))\ \%\ MOD$.
  - Trong đó ch là 1 kí tự tiếng anh in thường được viết thêm vào cuối xâu S.
  - $odr(ch)$ là thứ tự của kí tự ch trong bảng chữ cái tiếng anh. Ví dụ: $odr(a) = 1, odr(b) = 2, \ldots$
  - MOD là một số nguyên có dạng 2^M.

Phần nội dung của tờ giấy gồm 3 số nguyên N, K, M. Dựa vào nội dung và cách giải các nhà khảo cổ phải giải mã chúng bằng cách tìm ra xâu S gồm N kí tự sao cho $f(S) = K$.

Yêu cầu: Hãy tính xem có tất cả bao nhiêu xâu S thỏa mãn.

Subtask 1 ($1 \leq N \leq 5$): $O(26^N)$

Thực hiện xét tất cả các xâu và cập nhập kết quả.

Subtask 2 ($1 \leq N \leq 10; 6 \leq M \leq 15$): $O(26^{N/2} + 4^M)$

- Sử dụng kỹ thuật "meet-in-the-middle" bằng cách xét tất cả các xâu có độ dài $N/2$ và độ dài $N - N/2$.
- Gọi $cnt_1[K_1]$ là số lượng xâu S_1 độ dài $N/2$ thỏa mãn $f(S_1) = K_1$, $cnt_2[K_2]$ là số lượng xâu S_2 có độ dài $N - N/2$ thõa mãn $f(S_2) = K_2$.
- Xâu S được ghép từ xâu $S_1 + S_2$ thỏa mãn: $K_1 * 33^{N-N/2}$ xor $K_2 = K$ với $0 \leq K_1, K_2 < 2^M$.
  Khi đó số lượng xâu S được tính bằng $cnt_1[K_1] * cnt_2[K_2]$.

Subtask 3 ($1 \leq N \leq 10; 6 \leq M \leq 25$): $O(26^{N/2} + 2^M)$

Vì $M \geq 6$ nên $Mod \geq 32 > 26$, ta có nhận xét:

$$S' = ((S * 33)\ xor\ x)\ \%\ Mod$$
$$\leftrightarrow S' = (S * 33)\ \%\ Mod\ xor\ (x\ \%\ Mod)$$
$$\leftrightarrow S'\ xor\ x = (S * 33)\ \%\ Mod$$
$$\leftrightarrow (S'\ xor\ x) * inv(33, Mod) = S$$

Như vậy ta có 2 công thức xuôi và ngược như sau:

- $S' = ((S * 33)\ xor\ x)\ \%\ Mod$
- $S = (S'\ xor\ x) * inv(33, Mod)$

Gọi $cnt_1[K_1]$ là số lượng xâu S_1 có $f(S_1) = K_1$ được tính theo công thức xuôi; $cnt_2[K_2]$ là số lượng xâu S_2 có $f'(S_2) = K_2$ được tính theo công thức ngược.

Xâu S được ghép từ xâu $S_1 + S_2$ thỏa mãn nếu: $K_1 = K_2$ với $0 \leq K_1, K_2 < 2^M$. Khi đó số lượng xâu S được tính bằng $cnt_1[K_1] * cnt_2[K_2]$.

Bài 1. Giải mã – DECRYPT

Bài 2. Công thức nấu ăn – RECEIPT

Bài 3. Tom và Jerry – TOMJERRY

Công thức nấu ăn – Receipt

Công thức mà Kiên nghĩ ra là một dãy số gồm N số nguyên $a_1, a_2, \ldots, a_N$. Để làm cho món ăn trở nên ngon nhất thì Kiên phải cắt dãy số trên thành các phần khúc, và độ ngon của món ăn bằng tổng giá trị của các phân khúc. Giá trị của một phân khúc được xác định bằng chênh lệch giữa số lớn nhất và số nhỏ nhất trong phân đoạn đó.

Kiên có Q lần chỉnh sửa công thức. Ở mỗi lần chỉnh sửa, anh ấy sẽ tăng các giá trị của các số $a_l, a_{l+1}, \ldots, a_r$ lên x.

Yêu cầu: Hãy tính độ ngon tối đa của món ăn được tạo từ công thức sau mỗi lần chỉnh sửa.

Subtask 1 ($1 \leq N, Q \leq 200$): $O(Q * N^2)$

Đặt $Mx[i, j]$ là giá trị lớn nhất trong đoạn $a_i, a_{i+1}, \ldots, a_j$, $Mn[i, j]$ là giá trị nhỏ nhất trong đoạn $a_i, a_{i+1}, \ldots, a_j$.
Thực hiện quy hoạch động $dp[i]$ độ ngon lớn nhất khi xét đến số thứ i trong dãy.

- $dp[i] = \max(dp[i], dp[j] + Mx[j+1, i] - Mn[j+1, i])$ với $1 \leq j < i$.

Kết quả bài toán là $dp[N]$.

Subtask 2 ($1 \leq N \leq 3000$): $O(Q * N)$

- Khi chia dãy $a_1, a_2, \ldots, a_N$ thành các phân khúc thì mỗi phân khúc phải là một dãy tăng ngặt dài nhất có thể hoặc một dãy giảm ngặt dài nhất có thể. Với một phân khúc $a_l, a_{l+1}, \ldots, a_r$ thì giá trị của nó là
  $|a_l - a_r| = |a_l - a_{l+1}| + |a_{l+1} - a_{l+2}| + ... + |a_{r-1} - a_r|$.
- Đặt $d[i] = a_i - a_{i+1}$, giá trị của một phân khúc $a_l, a_{l+1}, \ldots, a_r$ bằng $|d_l| + |d_{l+1}| + ... + |d_{r-1}|$.
  Suy ra, kết quả của bài toán là tổng một số giá trị $|d_i|$, những giá trị $|d_i|$ không được tính trong kết quả đồng nghĩa với việc phần tử a_i và a_{i+1} nằm trong 2 phân khúc khác nhau. Tại những vị trí i thỏa mãn $a_{i-1} < a_i > a_{i+1}$ hoặc $a_{i-1} > a_i < a_{i+1}$ thì bắt buộc ta phải tách ra 2 phân khúc ở giữa 2 điểm $(i-1, i)$ hoặc $(i, i+1)$ điều đó có nghĩa chỉ 1 trong 2 giá trị $|d_{i-1}|$ hoặc $|d_i|$ được tính vào kết quả và khi đó $d_{i-1} * d_i < 0$.

Subtask 2 ($1 \leq N \leq 3000$): $O(Q * N)$

Gọi $dp[i, j]$ là độ ngon tối đa khi xét đến phần tử d_i.

- $j = 0$ nếu giá trị $|d_{i-1}|$ không được tính vào kết quả.
- $j = 1$ nếu giá trị $|d_(i - 1)|$ được tính vào kết quả.

Ta có công thức:

- $dp[i, 0] = \max(dp[i - 1, 0], dp[i - 1, 1])$.
- $dp[i, 1] = \max(dp[i - 1, 0], dp[i - 1, 1] * (d_{i-1} * d_i \geq 0)) + |d_i|$.

Kết quả bài toán: $\max(dp[N, 0], dp[N, 1])$.

Subtask 3 ($1 \leq N \leq 200000$): $O((N+Q) * \log N)$

Sử dụng cây Segment Tree, tại mỗi nút *id* quản lý đoạn $d_l, d_{l+1}, \ldots, d_r$ với 4 trường $(id.dp[0,0], id.dp[0,1], id.dp[1,0], id.dp[1,1])$ như sau:

- $ST[id].dp[i,j]$ là độ ngon tối đa trong đoạn $d_l, d_{l+1}, \ldots, d_r$ sao cho:
  - $i = 0$ nếu giá trị $|d_l|$ không được tính vào kết quả.
  - $i = 1$ nếu giá trị $|d_l|$ được tính vào kết quả.
  - $j = 0$ nếu giá trị $|d_r|$ không được tính vào kết quả.
  - $j = 1$ nếu giá trị $|d_r|$ được tính vào kết quả.
- Cập nhập nút *id* từ 2 nút con $2id$ và $2id+1$ lên như sau:
  - $ST[id].dp[i,j] = \max(ST[id].dp[i,j], ST[2id].dp[i,a] + ST[2id+1].dp[b,j])$ trừ trường hợp $a = b = 1$ và $d_{mid} * d_{mid+1} < 0$ (*mid* điểm thỏa mãn nút con bên trái quản lý đoạn $d_l, d_{l+1}, \ldots, d_{mid}$, nút con bên phải quản lý đoạn $d_{mid+1}, d_{mid+2}, \ldots, d_r$).

Bài 1. Giải mã – DECRYPT

Bài 2. Công thức nấu ăn – RECEIPT

Bài 3. Tom và Jerry – TOMJERRY

Tom và Jerry – TOMJERRY

- Có N đài phun nước và $N-1$ lối đi kết nối liên thông chúng. Ở đài phun nước thứ i hiện có p_i con chim bồ câu. Trong túi Jerry lúc này có v mẩu bánh mì. Nếu Jerry thả một mẩu bánh mỳ tại một đài phun nước thì tất cả các con chim bồ câu từ những đài phun nước lân cận khác cũng bay đến đài phun nước này.
- Đầu tiên, Jerry đến đài phun nước i và gặp p_i chim bồ câu. Sau đó, cậu ấy thả mẩu bánh mì và rời khỏi đó ngay lập tức. Những chú chim bồ câu từ những đài phun nước lân cận di chuyển đến đài phun nước i trước khi Jerry đến đài phun nước tiếp theo. Jerry có thể vào công viên ở bất kỳ đài phun nước nào, chạy qua một số lối đi (nhưng không bao giờ chạy qua lại cùng một lối đi hai lần), và sau đó rời khỏi công viên bất cứ nơi nào cậu ấy muốn. Sau khi Jerry thoát công viên, Tom sẽ đi vào và đi qua chính xác con đường mà Jerry đã đi.
- Jerry muốn tối đa hóa chênh lệch về số lượng chim bồ câu mình gặp và số lượng chim bồ câu mèo Tom gặp nếu cậu thả không quá v mẩu bánh mì. Lưu ý, chỉ có những con chim ở đài phun nước từ trước khi Jerry đến mới tính vào số lượng chim bồ câu Jerry gặp. Hãy xác định sự chênh lệch tối đa đó.

Subtask 1 ($N \leq 10$): $O(2^N * N^2)$

Ta xét tất cả các trường hợp thả mẩu bánh mỳ tại một số đài phun nước, tổng cộng có 2^N cách thả mẩu bánh mỳ như vậy. Số lượng đường trên đồ thị là N^2 nên độ phức là $O(N^2 * 2^N)$.

Subtask 2 ($N \leq 1000$): $O(N^2 * v \log v)$

- Giả sử Jerry đi đến đài phun nước v từ đài phun nước t và định đi đến đài phun nước w tiếp theo. Ta đặt $g(v) = \sum_{x \text{ lân cận } v} p(x) - p(t)$.
- Nếu Jerry di chuyển theo đường đi $x_1 \rightarrow x_2 \rightarrow ... \rightarrow x_k$ và thả mẩu bánh mỳ ở một số đài phun nước $x_{i_1}, x_{i_2}, ..., x_{i_c}$ thì chênh lệnh về số lượng chim bồ câu Tom gặp so với Jerry là $g(x_{i_1}) + g(x_{i_2}) + ... + g(x_{i_c})$.
- Ta có thể hiểu như sau: Khi Jerry từ đài phun nước t đến đài phun nước v. Nếu Jerry thả mẩu bánh mỳ ở v, thì những con chim bồ câu ở t cũng bay sang nhưng chúng đã gặp từ trước và đã tính vào số lượng của Jerry nên sẽ không tính thêm vào nữa. Vậy nếu việc chim bồ câu từ t bay sang v không có ý nghĩa gì cả, cho nên ta giả sử những con chim bồ câu từ t không bay sang v khi Jerry thả mẩu bánh mỳ ở v.

Subtask 2 ($N \leq 1000$): $O(N^2 * v \log v)$

- Vì thế mới đặt $g(v) = \sum_{x \text{ lân cận } v} p(x) - p(t)$.
- Ta có tổng cộng N^2 đường đi, với mỗi đường đi xuất phát từ x_1 và kết thúc tại x_k. Ta tính các giá trị $g(x_1), g(x_2), ..., g(x_k)$ và chọn ra v giá trị lớn nhất trong đó cộng lại. Trong quá trình DFS từ x_1 ta tính các giá trị $g()$ và cho vào set hoặc priority queue để dễ dàng lấy các giá trị lớn nhất.

Subtask 3 (Một đường đi tối ưu bắt đầu từ đài phun nước 1): $O(N * v * \log N)$

Làm tương tự subtask 2. Ở subtask này ta đã biết vị trí xuất phát là đài phun nước 1 nên cứ thực hiện DFS từ đó.

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

- Ta coi đỉnh gốc trong đồ thị này là đỉnh 1.
- Gọi $up[x, i]$ là chênh lệch lớn nhất của một đường đi xuất phát từ trong cây con gốc x và đi đến x với số lượng mẩu bánh mỳ tối đa đã sử dụng là i.
- Gọi $down[x, i]$ là chênh lệch lớn nhất của một đường đi qua x và kết thúc ở trong cây con gốc x với số lượng mẩu bánh mỳ tối đa đã sử dụng là i.
- Xét tại đỉnh x có đỉnh cha là par và các đỉnh con là $u_1, u_2, \ldots, u_k$. Đặt $Sum = p_{u_1} + p_{u_2} + ... + p_{u_k} + p_{par}$.

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

Duyệt lần lượt các con $u_1, u_2, ..., u_k$. Tại đỉnh con u_j ta có:

- Ta cập nhập kết quả: $Ans = \max(Ans, up[x, i] + down[u_j, v - i])$. Có thể hình dung rằng đường đi xuất phát từ một đỉnh bên ngoài cây con gốc u_j, sau đó đi đến x, rồi đi vào cây con gốc u_j.
- $up[x, i] = \max\{up[x, i-1], up[u_j, i], up[u_j, i-1] + Sum - p_{u_j}\}$.
- $down[x, i] = \max\{down[x, i-1], down[u_j, i], down[u_j, i-1] + Sum - p_{par}\}$.

Lưu ý ta phải thực hiện duyệt lại các con theo thứ tự $u_k, u_{k-1}, ..., u_1$ vì ta chỉ mới xét các con đường xuất phát từ trong cây con u_a đến trong cây con u_b với $a < b$. Nên ta phải thực hiện duyệt ngược lại theo thứ tự trên và quy hoạch động tương tự.