SOL DAY 09/10

Bài 1: Khiêu vũ: TKNP

Cơ bản

Bài 2: Khiêu vũ 2: TKNP

Bài toán: cho hai dãy số nguyên nguyên dương, $b_1, b_2, \dots, b_n$ và $g_1, g_2, \dots, g_m$ và Q truy vấn, mỗi truy vấn gồm hai số nguyên dương X_k, Y_k. Bài toán yêu cầu đếm có bao nhiêu cặp (i, j) sao cho $X_k \le b_i \times g_j \le Y_k$.

Thuật toán 1: Duyệt mọi cặp (i, j) và đếm có bao nhiêu cặp thỏa điều kiện $X \le b_i \times g_j \le Y$. Độ phức tạp $O(QNM)$. Thuật toán có thể thực hiện subtask 1.

Thuật toán 2: Bởi $Q \le 1000$, nên ta cần tính trước $N \times M$ tích lưu vào mãng p và sắp (không giảm) để các tích có giá trị liên tục, với mỗi truy vấn ta dùng tìm kiếm nhị phân để tìm phần tử đầu p_j và cuối p_k sao $X \le p_j$ và $p_k \le Y$. Độ phức tạp: $O(NMlog(NM))$ cho sắp xếp, tìm nhị phân $O(log(NM))$. Vậy độ phức tạp cho cả thuật toán là: $O((NM + Q)log(NM))$. Thuật toán có thể thực hiện subtask 1-2.

Thuật toán 3: Ta thấy rằng nếu sắp không giảm b_i, g_j thì khi ghép cặp lần lượt ta cũng được một dãy tích không giảm. Hay nói cách khác nếu xây dựng một bảng $N \times M$, mỗi ô (i, j) có giá trị là tích $b_i \times g_j$ của hai dãy đã sắp thì giá trị trong các ô của bảng sẽ không giảm từ trái sang phải và từ trên xuống dưới. Từ nhận xét đó, do các dòng của bảng đã được sắp nên với mỗi dòng ta sẽ dùng tìm kiếm nhị phân để tìm các tích thỏa mãn điều kiện và cộng thêm vào kết quả.

Vì bảng như mô tả là rất lớn nên ta chỉ thực hiện việc truy cập các phần tử của bảng thông qua cặp chỉ số (i, j) trên các dãy sau khi sắp. Độ phức tạp: $O(NlogN + MlogM)$ cho việc sắp các dãy, mỗi dòng ta thực hiện tìm nhị phân trên dãy M phần tử nên sẽ mất $O(QNlogM)$. Vậy độ phức tạp cho cả thuật toán là: $O(NlogN + MlogM + QNlogM)$. Thuật toán có thể thực hiện subtask 1-3.

Thuật toán 4: Cải tiến thuật toán 3.

Với mỗi truy vấn, từ thuật toán trên ta thấy với mỗi dòng i của bảng ta cần tìm hai chỉ số j và k. Trong đó j là vị trí đầu tiên sao cho $b_i \times g_j \ge X$ và k là vị trí đầu tiên sao cho $b_i \times g_k > Y$. Do g sắp không giảm nên $X \le b_i \times g_l \le Y$ khi và chỉ khi $j \le l < k$. Nói cách khác trên mỗi dòng chỉ có một đoạn gồm $k - j$ ô thỏa điều kiện cần tìm (có l chỉ số như vậy). Khi i tăng thì các đoạn này sẽ không dịch sang phải bảng hay j và k không lớn hơn các giá trị nhận được từ dòng trước đó. Do b cũng sắp không giảm nên nếu $b_i \times g_j \ge X$ thì $b_{i+1} \times g_j \ge X$, rỏ ràng giá trị j mới sẽ không thể lớn hơn giá trị j trước đó. Tương tự như vậy cho k.

Từ nhận xét trên, việc tìm j chỉ cần duyệt qua các giá trị giảm xuất phát từ giá trị j đã nhận trước đó. Với N dòng thì j chỉ dịch tối đa M lần. Tương tự cho k, nên khi này độ phức tạp của thuật toán xác định các tích thỏa mãn điều kiện trên N dòng chỉ còn là $O(N + M)$. Độ phức tạp chung cho cả thuật toán là $O(NlogN + MlogM + Q(N + M))$. Thuật toán có thể thực hiện subtask 1-4.

Bài 3. Hệ thống tưới nước: DP

Thuật toán:

Từ dữ liệu đã cho, xây dựng mảng D với D[i,j]=1 nếu ô (i,j) trồng cỏ, D[i,j]=0 nếu ô (i,j) không trồng cỏ,

Gọi C[i,j] là tổng số chi phí nhỏ nhất khi xây dựng một hình chữ nhật có kích thước: 1..i và 1..j. Ta sẽ có công thức truy hồi:

Nếu D[i,j]=1 (ô (i,j) trồng cỏ) thì:

+ Nếu hàng i và cột j không xây dựng hệ thống tưới (HTT) thì:

C[i,j]:=C[i-l,j-l] + Min {A[i], B[j]}

ngược lại C[i,j]:=Min{C[i- l,j]+A[i] , C[i,j-1]+B[j]}

Nếu D[i,j]=0 (ô (i,j) không trồng cỏ) thì:

+ Nếu hàng i không xây dựng HTT và cột j xây dựng HTT thì : C[i,j]:=C[i-l,j]

+ Nếu hàng i xây dựng HTT và cột j không xây dựng HTT thì : C[i,j]:=C[i,j-1]

+ Nếu hàng i và cột j xây dựng HTT thì: C[i,j]:=Max{C[i- l,j]+A[i], C[i,j-1] + B[j]}

+ Nếu hàng i và cột j không xây dựng HTT thì chia ra các trường hợp:

- D[i-l,j] =1 và D[i,j-1] =1
- D[i-l,j] =0 và D[i,j-1] =1
- D[i-l,j] =1 và D[i,j-1] =0
- D[i-l,j] =0 và D[i,j-1] =0