Ý tưởng giải bài ngày 11/05/2013

Đỗ Ngọc Khánh

Ngày 14 tháng 5 năm 2013

1 Subpair

1.1 Bài toán tìm một xâu con chung dài nhất của P và Q

Trong bài này ta xét các xâu con chung gồm các kí tự nằm liên tiếp trong một xâu.

Gọi $F(i,j)$ là độ dài xâu con chung dài nhất kết thúc tại P_i và Q_j.

- Nếu $P_i = Q_j$ thì $F(i,j) = F(i-1,j-1)+1$.
- Ngược lại: $F(i,j) = 0$.

Tương tự cách xây dựng F ta có thể xây dựng hàm $G(i,j)$ với P_i, Q_j là điểm bắt đầu của xâu con chung.

1.2 Bài toán tìm hai xâu con chung phân biệt có tổng độ dài dài nhất của P và Q

Nhận thấy rằng ta luôn tìm được hai vị trí x và y ở xâu P và Q, chia mỗi xâu làm hai phần sao cho $P_{1..x}$ và $Q_{1..y}$ chứa a, $P_{x+1..|P|}$ và $Q_{y+1..|Q|}$ chứa b. Với mỗi vị tri ngắt (x,y) thì độ dài tối đa của hai xâu (a,b) cần tìm là $\max F(i,j) + \max G(s,t)$ với $1 \leq i \leq x < s \leq |P|$, $1 \leq j \leq y < t \leq |Q|$.

Cách cài đặt như trên có độ phức tạp $O\left((|P||Q|)^2\right)$ là quá lớn nhưng nếu ta tính sẵn $maxF(i,j) = \max\{F(i,j-1), F(i-1,j), F(i,j)\}$ và lưu lại một cặp (u,v) sao cho $maxF(i,j) = F(u,v)$ để truy vết sau này, tương tự với hàm G thì độ phức tạp thuật toán chỉ còn $O(|P||Q|)$.

2 Fibsseq

1. Nếu dãy có độ dài $m = 1$ thì kết quả bài toán là 1.

2. Nếu dãy có độ dài $m > 1$

   Gọi $F(u, v)$ là độ dài của dãy con dạng Fibonacci dài nhất có phần tử cuối cùng ở vị trí v và phần thử gần cuối ở vị trí u. Nếu như tồn tại $1 \leq k < u$ sao cho $a_k = a_v - a_u$ thì $F(u, v) = \max\{F(u, v), F(k, u) + 1\}$, lưu $T(u, v) = k$ để truy vết. Nếu không tìm được k thì $F(u, v) = 2$.

   Cách cài đặt không tối ưu thêm có độ phức tạp $O(m^3)$. Nhưng ta nhận thấy rằng nếu như tồn tại $k' < k$ và $a_{k'} = a_k$ thì $F(k', u) \leq F(k, u)$ (vì chúng ta có thể thay vì chọn phần tử $a_{k'}$ mà chọn a_k thì độ dài dãy sẽ không giảm) nên giá trị k cần chọn luôn lớn nhất có thể. Bằng các lưu vào `map` các vị trí xuất hiện của một giá trị trong dãy, thời gian tìm kiếm k sẽ là $O(\log m)$.

   Tóm lại bài toán có lời giải $(m^2 \log m)$.

3 Longdom

3.1 Bài toán phủ sử dụng domino 2×1

Không mất tính tổng quát ta có thể giả sử số cột m của bảng không lớn hơn số hàng n.

Để không phải lưu lại trạng thái của cả bảng, ta quy ước thứ tự phủ là phải đặt domino phủ kín được hàng r thì mới bắt đầu đặt domino phủ kín hàng $r+1$. Trên cùng một hàng ta tiến hành phủ từ trái sang phải. Với cách phủ như trên thì bảng luôn luôn có một số hàng đã phủ kín, rồi đến một hàng đang được phủ gây ảnh hưởng tới hàng ngay sau nó và các hàng còn lại hoàn toàn chưa được phủ.

Hàng r có trạng thái phủ s nếu như cột c của hàng r chưa được phủ thì bit c của s là bit 1, ngược lại bit c của s là bit 0.

Gọi $F(r, s_1, s_2)$ là số cách phủ kín $r - 1$ hàng đầu của bảng, hàng r đang được phủ, mang trạng thái s_1, hàng $r + 1$ mang trạng thái s_2.

- Nếu s_1 cho thấy rằng hàng r đã được phủ kín thì $F(r + 1, s_2, 0) = F(r + 1, s_2, 0) + F(r, s_1, s_2)$.

- Nếu s_1 cho thấy rằng hàng r chưa được phủ kín, ta chọn ô đầu tiên chưa được phủ, thử đặt domino theo cả hai hướng, khi đó hai hàng r và $r+1$ có trạng thái mới là s'_1 và s'_2, $F(r, s'_1, s'_2) = F(r, s'_1, s'_2) + F(r, s_1, s_2)$.

Theo như cách phủ ta quy ước thì không bao giờ có một ô ở hàng $r+1$ được phủ mà ô cùng cột ở hàng r chưa được phủ, tức là xét 1 cột trên 2 hàng liên tiếp chỉ có 3 trạng thái phủ cho 2 hàng này. Như vậy thì lời giải trên có độ phức tạp $O(n.3^m)$.

Cơ sở: $F(1, 0, 0) = 1$. Kết quả bài toán: $F(n+1, 0, 0)$.

3.2 Bài toán phủ sử dụng domino 3×1

Ta vẫn quy ước cách phủ như trên. Theo cách phủ này thì ta chỉ cần quan tâm tới trạng thái của 3 hàng cuối cùng.

Gọi $F(r, s_1, s_2, s_3)$ là số cách phủ kín domino $r-1$ hàng đầu tiên, ba hàng $r, r+1, r+2$ tương ứng mang các trạng thái s_1, s_2, s_3.

- Nếu s_1 cho thấy hàng r đã được phủ kín thì $F(r+1, s_2, s_3, 0) = F(r+1, s_2, s_3, 0) + F(r, s_1, s_2, s_3)$.

- Nếu s_1 cho thấy hàng r chưa được phủ kín, ta thử phủ kín ô đầu tiên chưa được phủ theo cả hai cách đặt domino, ba hàng $r, r+1, r+2$ tương ứng mang các trạng thái mới là s'_1, s'_2, s'_3: $F(r, s'_1, s'_2, s'_3) = F(r, s'_1, s'_2, s'_3) + F(r, s_1, s_2, s_3)$.

Tương tự với domino 2×1, ta thấy rằng với 1 cột trên 3 hàng liên tiếp chỉ có 4 cách phủ cho 3 hàng này. Như vậy lời giải trên có độ phức tạp $O(n.4^m)$.

Cơ sở: $F(1, 0, 0, 0) = 1$. Kết quả bài toán: $F(n+1, 0, 0, 0)$.

4 Orderenc

4.1 Giới hạn dễ: $n \leq 200$

Gọi $F(L, R)$ là số bit tối thiểu cần sử dụng để mã hoá các từ từ L đến R. Dễ thấy rằng ta không bao giờ để cho tất cả các xâu mã hoá bắt đầu bằng cùng một bit 0 hoặc bit 1, nên luôn tồn tại một điểm cắt k $(L \leq k < R)$ sao cho các từ từ L đến k được mã hoá bắt đầu bằng bit 0, các từ từ $k+1$ đến R được mã hoá bắt đầu bằng bit 1. Vậy thì: $F(L, R) = F(L, k) + F(k+1, R) + \sum_{L \leq i \leq R} p_i$. Lí do như vậy là vì tổng tần suất của các từ bằng bao nhiêu thì ta tốn thêm bấy nhiêu bit đưa vào đầu phần mã hoá các từ, hai phần sau phân chia độc lập với nhau. Việc ta cần làm là tìm k sao cho $F(L, R)$ tính theo công thức trên là nhỏ nhất có thể, lưu $T(L, R) = k$ để truy vết sau này.

Nhận thấy rằng $F(a, a) = 0$, nếu chỉ có 1 xâu thì ta mã hoá nó bằng 0 bit.

Lời giải này có độ phức tạp $O(n^3)$.

4.2 Giới hạn khó: $n \leq 2000$