Ý tưởng giải bài ngày 15/05/2013

Đỗ Ngọc Khánh

Ngày 15 tháng 5 năm 2013

1 Ncontest

Nhận thấy rằng để số bài giải được là tối đa ta sẽ giải những bài dễ nhất cho đến khi hết giờ (để tiện cho việc trình bày, ta coi bài giải hết ít thời gian hơn là dễ hơn).

Để cho thời gian phạt là ít nhất có thể ta sẽ giữ lại tất cả những bài giải xong trước 0 giờ, chờ đến đúng thời điểm 0 giờ mới nộp, hơn nữa ta luôn giải bài dễ trước bài khó. Thật vậy, giả sử ta không giải bài theo thứ tự trên thì chắc chắn sẽ tồn tại hai bài giải liên tiếp i và j sao cho $a_i > a_j$. Xét các trường hợp có thể xảy ra:

1. Nếu cả hai bài được giải xong trước mốc 0 giờ thì ta đổi ngược lại giải bài j trước bài i không làm thay đổi thời gian phạt.

2. Nếu cả hai bài được bắt đầu giải T phút sau mốc 0 giờ, khi đó:

   - Nếu giải bài i trước thì thời gian phạt là: $(T + a_i) + (T + a_i + a_j)$.
   - Nếu giải bài j trước thì thời gian phạt là: $(T + a_j) + (T + a_j + a_i)$.

   Do $a_i > a_j$ nên giải bài j trước bài i cho thời gian phạt nhỏ hơn.

3. Nếu bài i được giải từ trước mốc 0 giờ và được hoàn thành tại phút T sau mốc 0 giờ thì:

   - Nếu giải bài i trước thì thời gian phạt là: $T + (T + a_j)$.
   - Nếu giải bài j trước thì thời gian phạt là: $\max\{0, T - a_i + a_j\} + (T + a_j)$.

   Do $a_i > a_j$ nên giải bài j trước bài i cho thời gian phạt nhỏ hơn.

4. Nếu bài i được giải xong trước mốc 0 giờ và bài j được giải hoàn thành tại phút T sau mốc 0 giờ thì ta đổi ngược lại giải bài j trước bài i thời gian phạt vẫn giữ nguyên.

Vậy ta đã chứng minh được thứ tự giải bài như trên luôn cho thời gian phạt nhỏ nhất.

Độ phức tạp của lời giải này phụ thuộc chủ yếu vào việc sắp xếp độ khó của bài. Với $n \leq 100$, tuỳ vào thuật toán sắp xếp được sử dụng mà ta có thể có lời giải với các độ phức tạp $O(n + \max a_i)$, $O(n \log n)$ hoặc $O(n^2)$.

2 Ticket

Để việc đếm lượng người được soát vé không bị lặp, ta sẽ đếm như sau: Giả sử các trạm soát vé được đặt tại các vị trí $x_1, x_2, \ldots, x_K$ (trạm soát vé đặt tại x sẽ soát vé những người đi trên chặng nối hai ga x và $x + 1$) thì ta đếm lượng người được soát vé tại trạm i là lượng người xuất phát từ $s \leq p_i$ và đi tới $t \leq p_{i+1}$, coi $x_{K+1} = N$. Với cách đếm như trên thì thứ tự đặt trạm soát vé tiện lợi cho việc tính toán nhất là đặt từ phải sang trái.

Gọi $f(n, k)$ là số khách tối đa được soát vé nếu như chỉ có n nhà ga đầu tiên và được đặt k trạm soát vé. Vậy thì:

- $f(n, 0) = 0$.
- Với $k > 0$, giả sử trạm soát vé thứ k được đặt tại vị trí c $(1 \leq c < n)$ thì ta cập nhật: $f(n, k) = \max\left\{f(n, k), f(c, k-1) + \sum_{s=1}^{c} \sum_{t=c+1}^{n} p_{s,t}\right\}$.

Nếu như không tối ưu gì thêm, lời giải trên có độ phức tạp $O(N^4)$ là quá lớn. Ta sẽ tối ưu lời giải trên như sau: Giả sử đặt trạm soát vé ở vị trí c thì khi lùi trạm soát vé này về vị trí $c - 1$, ta không soát được vé của những người bắt đầu đi từ c và đi không quá n nữa, nhưng bù lại ta soát được vé của những người dừng lại tại c, bằng một vài thao tác tiền xử lí đơn giản (trong thời gian $O(N^2)$) ta có thể tính được lượng khách được soát vé mới trong thời gian $O(1)$. Vậy thì lời giải trên được tối ưu xuống còn $O(N^3)$ là phù hợp với giới hạn bài toán.

3 Lollipop

Để việc trình bày được dễ dàng, ta phát biểu lại bài toán như sau: Cho một dãy số A có N phần tử, giá trị của mỗi phần tử là 1 hoặc 2. Hãy xác định một đoạn con của dãy A có tổng giá trị đúng bằng k cho trước. có M câu hỏi như vậy.

Đầu tiên ta tính giá trị của tiền tố i: $S_i = \sum_{k=1}^{i} A_i$ là tổng giá trị các số thuộc tiền tố độ dài i của A. Nếu có một tiền tố i có giá trị là v, ta lưu $p_v = i$, ngược lại $p_v = \infty$. Ta có một nhận xét như sau: Với $k < S_N$ nếu $p_k = \infty$ thì $p_{k+1} \neq \infty$, việc chứng minh nhận xét này là khá đơn giản.

Bây giờ ta sẽ đi tìm ra câu trả lời cho một truy vấn $k \leq S_N$ (hiển nhiên nếu $k > S_N$ thì không có đoạn con nào như vậy):

- Nếu $p_k \neq \infty$ thì câu trả lời là đoạn con $[1, p_k]$.
- Nếu $p_k = \infty$, đặt $v = p_{k+1} \neq \infty$ (theo nhận xét trên). Sau đó ta tiến hành dịch đầu, cuối của đoạn $[1, v]$ này cho tới khi được tổng đúng bằng k hoặc chỉ ra là không tồn tại đoạn như vậy. Giờ ta sẽ xét xem nếu như dịch đoạn $[i, j]$ có tổng bằng $k + 1$ sang phải thì nó sẽ dừng lại khi nào:
  - Nếu $A_i = 1$, ta dịch đoạn $[i, j]$ thành $[i + 1, j]$ và thu được kết quả bài toán.
  - Nếu $A_i = 2$:
    * Nếu $A_{j+1} = 1$, ta dịch đoạn $[i, j]$ thành $[i + 1, j + 1]$ và thu được kết quả bài toán.
    * Nếu $A_{j+1} = 2$, ta dịch đoạn $[i, j]$ thành $[i + 1, j + 1]$ và thu được một đoạn mới tương đương với đoạn cũ, ta sẽ tiếp tục dịch.

  Từ những phân tích trên ta có kết luận như sau: Nếu đoạn $[i, j]$ có tổng bằng k thì hoặc $A_{i-1} = 1$, hoặc $A_j = 1$. Như vậy ta sẽ xác định xem số 1 trong dãy gần phần tử A_1 nhất về bên phải là ở đâu rồi thử xem vị trí sau nó có thể là vị trí bắt đầu của đoạn con cần tìm hay không. Nếu không được thì ta lại tìm số 1 trong dãy gần phần tử A_v nhất về bên phải là ở đầu rồi thử xem vị trí đó có thể là vị trí kết thúc của đoạn con cần tìm hay không. Nếu vẫn không thoả mãn thì ta kết luận rằng không có đoạn con nào có tổng đúng bằng k.

  Việc thử và kiểm tra có thể thực trên trong thời gian $O(1)$ nếu như ta tính trước một mảng o trong thời gian $O(N)$, ở đó $o_i = j$ nếu j là số nhỏ nhất thoả mãn $j \geq i$ và $A_j = 1$, nếu không tìm được ta cho $j = \infty$. Vậy thì ta mất $O(M)$ để trả lời M câu hỏi, $O(N)$ để khởi tạo. Tóm lại, bài toán có lời giải với độ phức tạp $O(N + M)$.

4 Randl

4.1 Bài toán 1: Khu vườn có dạng đường thẳng, ô đất 1 không kề với ô đất N

Ta có một số nhận xét như sau:

1. Ta không bao giờ bắn súng vào hai ô đất liên tiếp, bởi vì khi ta bắn vào một ô thì thỏ ở hai ô kề với nó sẽ nhảy đi mất, nếu ta tiếp tục bắn thì sẽ là bắn vào ô trống, không đuổi được con thỏ nào.

2. Nếu như được bắn $k > 0$ viên đạn thì ta đuổi được hết số thỏ nằm ở $2k-1$ ô liên tiếp trong vườn (cách một ô bắn một viên) và ta có thể bắn k viên đạn này theo bất kì thứ tự nào.

3. Theo cách bắn trên, một con thỏ nếu có nhảy thì chỉ nhảy được sang những ô có chênh lệch về vị trí không quá 1 đơn vị số với vị trí ban đầu.

Trong hàm quy hoạch động, ta sẽ tiến hành bắn súng từ phải sang trái. Gọi $f(n, k, c)$ là số thỏ tối đa có thể đuổi được nếu như vườn có n ô đất từ 1 đến n, còn k viên đạn để bắn, $c = 0$ nếu như thỏ ở ô đất $n+1$ không nhảy sang ô đất n, ngược lại $c = 1$. Vậy thì:

- Nếu $k = 0$ thì $f(n, k, c) = 0$.

- Nếu $k > 0$, xét hai trường hợp:

    1. Không bắn súng vào ô đất n, như vậy thỏ ở ô đất n sẽ không nhảy sang ô đất $n-1$, ta đuổi được $f(n-1, k, 0)$ con thỏ.
    2. Bắn súng vào ô đất n, như vậy thỏ ở ô đất n sẽ bị đuổi, có thể bao gồm cả thỏ ban đầu ở ô $n+1$, toàn bộ thỏ ở ô đất $n-1$ sẽ nhảy sang ô đất $n-2$, ta đuổi được $f(n-2, k-1, 1) + a_n + \begin{cases} a_{n+1} & \text{nếu } c = 1 \\ 0 & \text{nếu } c = 0 \end{cases}$ con thỏ.

    Do lượng thỏ cần đuổi đi phải là lớn nhất có thể nên $f(n, k, c)$ sẽ nhận giá trị lớn hơn trong hai giá trị trên.

Kết quả bài toán là $f(N, K, 0)$ có thể tính trong thời gian $O(NK)$.

4.2 Bài toán 2: Khu vườn có dạng đường tròn, ô đất 1 kề với ô đất N

Sau khi giải xong bài toán trên ta có thể nghĩ ngay tới việc thử N điểm cắt khác nhau để duỗi đường tròn này thành đường thẳng, nhưng thực tế ta chỉ cần xét ba trường hợp là đủ.

4.2.1 Trường hợp 1: Cả hai ô đất 1 và N đều không bị bắn vào

Khi đó ta có thể duỗi khu vườn tròn này thành khu vườn thẳng gồm $N-2$ ô đất từ ô đất 2 đến ô đất $N-1$, số viên đạn giữ nguyên là K viên. Cách xử lí trường hợp này rất đơn giản.

4.2.2 Trường hợp 2: Ô đất 1 bị bắn vào nhưng ô đất N thì không

Do thứ tự bắn các viên đạn là không quan trọng nên ta có thể duỗi khu vườn thành một khu vườn mới gồm N ô đất có số thỏ như sau: 0, 0, $(a_2 + a_3)$, a_4, $\ldots$, $(a_{n-1} + a_n)$, 0. Sau đó ta tính số thỏ tối đa đuổi được trên khu vườn mới này với số viên đạn còn lại là $K-1$. Kết quả của trường hợp này phải cộng thêm lượng thỏ ban đầu ở ô 1.

4.2.3 Trường hợp 3: Ô đất N bị bắn vào nhưng ô đất 1 thì không

Trường hợp này có thể giải quyết tương tự như trường hợp trên.

Tóm lại

Với mỗi trường hợp ta quy hoạch động trong thời gian $O(NK)$, có ba trường hợp khác nhau nên độ phức tạp của lời giải này vẫn là $O(NK)$.