Solution SEQ23

Vấn đề đáng bàn ở đây là subtask 2 của bài toán. Trước hết, với mỗi vị trí i trên dãy số ta cần xác định hai giá trị $L[i]$ và $R[i]$ thỏa mãn:

- $1 \leq L[i] \leq i \leq R[i] \leq N$;
- $a_i = max\{a_{L[i]}, a_{L[i]+1}, ..., a_{R[i]}\}$;
- $(R[i] - L[i])$ đạt giá trị lớn nhất.

Muốn xây dựng được mảng $L[i]$ và $R[i]$ với độ phức tạp $O(N)$ bạn phải có kiến thức về hàng đợi để tịnh tiến, ở đây mình xin chia sẻ một bài viết rất hay về vấn đề này trên VNOI wiki, nếu bạn đọc chưa biết thì hãy tham khảo trước khi tiếp tục quay trở lại SEQ23: `http://vnoi.info/wiki/algo/data-structures/deque-min-max`.

Chúng ta cần có một mảng B quản lý cặp số $(a_i, R[i] - L[i] + 1)$ trong đó a_i là phần tử $first$. Sau đó chúng ta sắp xếp mảng B tăng dần. Với mỗi truy vấn, chúng ta chặt nhị phân tìm vị trí p lớn nhất mà $B[p].first \leq K$, sau đó kết quả chính là $max\{B[1].second, B[2].second, ..., B[p].second\}$. Không khó để giải thích cho thuật toán này, khi mà một dãy số gồm các phần tử không lớn hơn K tương đương với việc phần tử lớn nhất của dãy đó phải nhỏ hơn hoặc bằng K. Sau đó thì với mỗi a_i ta lưu lại các dãy dài nhất có thể mà a_i là giá trị lớn nhất của dãy đó. Muốn tìm kiếm nhanh hơn thì chúng ta cần sắp xếp a_i lại rồi chặt nhị phân như trên, chú ý việc lấy $max\{B[1].second, B[2].second, ..., B[p].second\}$ nhanh ta có thể quy hoạch động từ trước nên mỗi truy vấn chỉ mất độ phức tạp $O(logN)$. Cuối cùng độ phức tạp bài toán chính là $O((Q+N)logN)$.