

Phương pháp: dùng binary index tree (BIT).

Ban đầu ta sắp xếp các phần tử của mảng theo thứ tự tăng dần (có lưu thứ tự của các phần tử). Sau đó dùng mảng place xác định vị trí của từng phần tử sau khi sắp xếp.

Sử dụng k cây BIT, BIT thứ x dùng để đếm số dãy nghịch thế có độ dài x. Mỗi đoạn $[i..j]$ dùng để tính số dãy nghịch thế từ $[a[i]..a[j]]$ (mảng sau khi đã sắp xếp).

Ta duyệt dãy theo thứ tự ban đầu nhờ vào mảng place. Với mỗi $a[place[i]]$, ta tìm số dãy nghịch thế độ dài từ 1 đến k kết thúc tại $a[place[i]]$. Vì trong dãy nghịch thế không tồn tại hai phần tử bằng nhau nên với các phần tử có giá trị bằng nhau, trong BIT ta cập nhật giá trị trên cùng một nút (trên nút bằng vị trí phần tử đầu tiên có giá trị bằng $a[place[i]]$). Ta tính số dãy nghịch thế độ dài x ($1 \leq x \leq k$) dựa vào số dãy nghịch thế độ dài x-1 và có chỉ số nằm trước $a[place[i]]$ (truy vấn trong cây BIT thứ x-1 trong đoạn $[1..v]$ vị trí đầu tiên có giá trị bằng $a[place[i]]-1$).

Kết quả bài toán là số dãy nghịch thế trong đoạn $[1..n]$ của cây BIT thứ k.

Độ phức tạp thuật toán: $n \log(n)$

Tham khảo lời giải các bài khác hoặc thảo luận ngay tại đây: <https://icnhoukdsiih.blogspot.com/>