Subtask #1 (cnt[k] ≥ 1e6):

- Vì mọi hoán vị đều có độ đẹp là val[k] nên đáp án là val[k] * n!.

Subtask #2 và #3 :

- Để giải quyết 2 subtask sau, chúng ta đi tính contribution của các giá trị val[i] vào tổng hay nói cách khác là cần tính xem với mỗi số i (1 ≤ i ≤ k) thì tồn tại bao nhiêu hoán vị có độ đẹp = val[i].

- Gọi f(i) là số hoán vị có độ đẹp ≤ val[i], theo phương pháp loại trừ sẽ có đúng f(i)-f(i-1) hoán vị có độ đẹp = val[i], vậy nếu ta tính được f(i) (1 ≤ i ≤ k) trong O(1) thì cả bài sẽ được giải quyết với độ phức là tạp là O(k).

- Cách tính f(i) (1 ≤ i ≤ k) trong O(1) :

+) Khi giả quyết các bài tập liên quan đến trung vị, có 1 trick cổ điển như sau : Cho mảng a[1…N] và 1 số x bất kỳ, ta dựng mảng b[1…N] sao cho b[i] = 1 nếu a[i] ≤ x và b[i] = -1 nếu a[i] > x, Khi đó nếu subarray b[l…r] của mảng b có **tổng các phần tử** ≥ 0 thì tương đương với việc subarray của mảng a[l…r] của mảng a có **median** ≤ x. Chúng ta sẽ lợi dụng tính chất này để xây dựng công thức tính f(i).

+) Để xây dựng công thức tính f(i) cần giải quyết bài toán sau : gọi c1 là số phần tử ≥ val[i] và c2 là số phần tử ≤ val[i] có trong mảng a (c1+c2 = N), ta cần tính xem dựng được bao nhiêu mảng b có n phần tử (trong đó bao gồm **c1 số 1** và **c2 số -1**) thỏa mãn không tiền tố nào của mảng b có tổng < 0. Vì các mảng b chỉ gồm 1 và -1, ta có thể coi b như là một dãy ngoặc với **c1 lần mở ngoặc** là **c2 lần đóng ngoặc.**

Subtask #2 (c1 = c2) : tương đương việc đếm số dãy ngoặc đúng, đáp án là số catalan thứ n.

Subtask #3 (c1, c2 là các số bất kỳ) : bài toán tổ hợp này có thể giải quyết trong O(1) bằng kỹ thuật catalan convolution được giới thiệu trong blog sau [Tutorial] Catalan Numbers and Catalan Convolution - Codeforces.

+) Gọi b(i) là số mảng b thỏa mãn với mỗi giá trị val[i] (1 ≤ i ≤ k) tương ứng, dễ thấy có c1! * c2! số hoán vị thỏa mãn cấu hình của mỗi mảng b, vậy ta có công thức sau : f(i) = b(i) * c1! * c2!.