Byron's solution:

1. Given [2, 5, 3, 4, 1]

2. sort them and give each number an index.

$$\Rightarrow \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ [1] & [2] & [3] & [4] & [5] \end{array}$$

☆ # of rating on the left of i and < ratings[i]

||

3. Compute less[i] using Binary Indexed Tree

for i, r in ratings:

and on the left

\# ratings < r

less[i] = $\boxed{\text{query(map[r], bit)}}$

update(map[r], bit, 1)

4. compute triplet[i] using BIT and less[i]

for i, r in ratings:

triplet[i] = query(map[r], bit, 2)

update(map[r], bit2, less[i])

for every r_j where $j < i$
and $r_j < r_i$
we sum over their
less[j]

Intuition:

$r_0\ r_1\ r_2\ r_3\ r_4\ r_5\ r_6\ r_7\ r_8$

less: [0] [1] [2] [3] [4] [5] [6] [7] [8]

When i == 5,
we have only
seen these numbers

Ex. triplet[5] =

sum(less[j] for $j < i$ and $r_j < r_i$)

//

So query(map[r_5], bit2)

Ensure $r_j < r_i$

Generalized to $K \geq 4$ soldiers:

Given: $[r_0\ r_1\ r_2\ r_3\ r_4\ r_5\ r_6\ r_7\ r_8]$

Compute triplet,

$$\text{triplet}[i] = \text{sum}(1 \text{ for } k<j<i \text{ and } r_k<r_j<r_i)$$

$$\text{quad}[i] = \text{Sum}(\text{triplet}[j] \text{ for } j<i \text{ and } r_j<r_i)$$

Ex.

$$[r_0\ r_1\ r_2\ r_3\ r_4\ r_5\ r_6\ r_7\ r_8]$$

If only $r_2 < r_6$, $r_4 < r_6$:

then $\text{quad}[6] = \text{triplet}[2] + \text{triplet}[4]$