# 1395. Count Number of Teams

There are `n` soldiers standing in a line. Each soldier is assigned a unique `rating` value.

You have to form a team of 3 soldiers amongst them under the following rules:

- Choose 3 soldiers with index (`i`, `j`, `k`) with rating (`rating[i]`, `rating[j]`, `rating[k]`).
- A team is valid if: (`rating[i] < rating[j] < rating[k]`) or (`rating[i] > rating[j] > rating[k]`) where (`0 <= i < j < k < n`).

Return the number of teams you can form given the conditions. (soldiers can be part of multiple teams).

Example 1:

```
Input: rating = [2,5,3,4,1]
Output: 3
Explanation: We can form three teams given the conditions. (2,3,4),
(5,4,1), (5,3,1).
```

Example 2:

```
Input: rating = [2,1,3]
Output: 0
Explanation: We can't form any team given the conditions.
```

Example 3:

```
Input: rating = [1,2,3,4]
Output: 4
```

Constraints:

- `n == rating.length`
- `3 <= n <= 1000`
- `1 <= rating[i] <= 105`
- All the integers in `rating` are unique.

```
def numTeams(self, rating: List[int]) -> int:
        ascending = 0
        descending = 0
```

```
        for i,ele in enumerate(rating):
            leftS,rightG,leftG,rightS = 0,0,0,0
            for el in rating[:i]:
                if el<ele:
                    leftS = leftS+1
                else:
                    leftG = leftG+1
            for el_ in rating[i+1:]:
                if el_<ele:
                    rightS = rightS+1
                else:
                    rightG = rightG+1
            ascending = ascending+leftS*rightG
            descending = descending+leftG*rightS
        return descending+ascending
```

The subset can also give answer but it was giving TLE.