Foundation Technical Training

Minimum Moves and Arrange Permutation

Name: Niranjan Kolpe, Batch: C#-Batch 2

Problem Statement 1:
"Minimum Moves"

Consider a chessboard of size N*N. On this board, exactly N queens are placed. An arrangement is valid if there is exactly one queen in one column (row contains any number of queens(up to N)).

Below are examples of valid and invalid arrangements:

- Valid arrangement – each column contains exactly one queen.
- Invalid arrangement – 3rd column contains 2 queens which violate our condition.

You may know that a queen can move horizontally, vertically, or diagonally in a game of chess to attack others. A valid arrangement is good if no queen attacks any other queen in the next move.

For rearrangement of queens, consider one move as moving a queen to one cell up or one cell down from its current position (without leaving the chessboard).

You cannot move the queen to another column. You are given valid arrangements.Find the minimum moves to make it good.

Function Description

In the provided code snippet, implement the provided minimumMoves(...) method using the variables to print the minimum moves. You can write your code in the space below the phrase "WRITE YOUR LOGIC HERE".

There will be multiple test cases running so the Input and Output should match exactly as provided.

Input Format

The first line contains N denoting the parameter for the size of the chessboard(N x N).

For the next N lines, each line contains a string of size N.

""Q"" denotes that queen is placed and ""."" denotes the empty cell.

Sample Input

4

.Q..

Q..Q

....

..Q.

Constraints

4 <= N <= 8

Output Format

A single line of output should contain minimum moves so that no queen attacks each other.

Sample Output

1

Explanation

Given Configuration:

. Q . .

Q . . Q

. . . .

. . Q .

By moving the queen of 2nd row, 1st column to 3rd row, 1st column we get the configuration as follows:

. Q . .

. . . Q

Q . . .

. . Q .

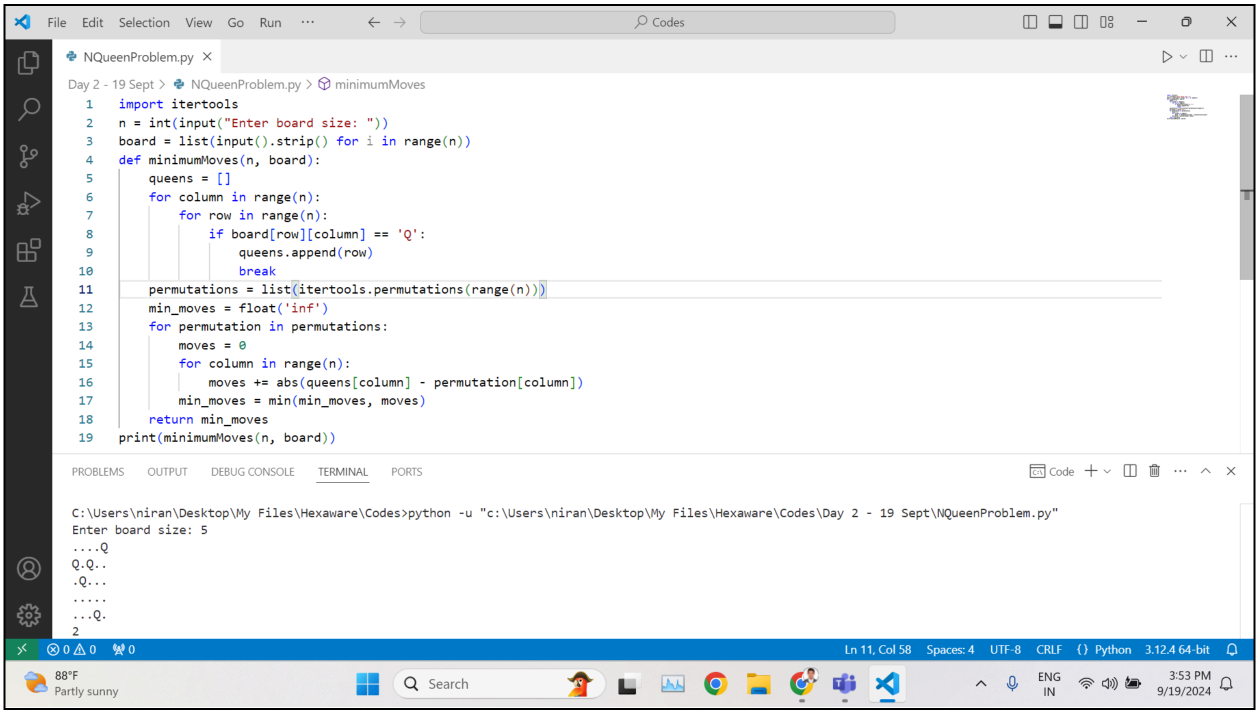
This configuration is good as no queen can attack another.

It is achieved with one move, and hence the output is 1."

Program Code:

```
import itertools

n = int(input("Enter board size: "))

board = list(input().strip() for i in range(n))

def minimumMoves(n, board):

    queens = []

    for column in range(n):

        for row in range(n):

            if board[row][column] == 'Q':

                queens.append(row)

                break

    permutations = list(itertools.permutations(range(n)))

    min_moves = float('inf')

    for permutation in permutations:

        moves = 0

        for column in range(n):

            moves += abs(queens[column] - permutation[column])

        min_moves = min(min_moves, moves)

    return min_moves

print(minimumMoves(n, board))
```

Output:


File Edit Selection View Go Run
Codes
NQueenProblem.py
Day 2 - 19 Sept > NQueenProblem.py > minimumMoves
import itertools
n = int(input("Enter board size: "))
board = list(input().strip() for i in range(n))
def minimumMoves(n, board):
    queens = []
    for column in range(n):
        for row in range(n):
            if board[row][column] == 'Q':
                queens.append(row)
                break
    permutations = list(itertools.permutations(range(n)))
    min_moves = float('inf')
    for permutation in permutations:
        moves = 0
        for column in range(n):
            moves += abs(queens[column] - permutation[column])
        min_moves = min(min_moves, moves)
    return min_moves
print(minimumMoves(n, board))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
C:\Users\niran\Desktop\My Files\Hexaware\Codes>python -u "c:\Users\niran\Desktop\My Files\Hexaware\Codes\Day 2 - 19 Sept\NQueenProblem.py"
Enter board size: 5
....Q
Q.Q..
.Q...
.....
...Q.
2
Ln 11, Col 58 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit
88°F
Partly sunny
Search
ENG
IN
3:53 PM
9/19/2024

Problem Statement 2:

"Arrange Permutation"

"Given an array, consisting of permutation of numbers till N i.e {1,2,3,....... N}.

An array element A[i] is considered local minima in an array if A[i-1] > A[i] < A[i+1]

An array element A[i] is considered local maxima in an array if A[i-1] < A[i] > A[i+1]

Check if it is possible to arrange the given permutation in any such way such that it consists of exactly the P number of maximas and exactly Q number of minimas.

If any such permutation is possible print ""Yes"", else print ""No"".

Function Description

In the provided code snippet, implement the provided checkPermutation() method using the variables to find if any such permutation is possible to print ""Yes"", else print ""No"". You can write your code in the space below the phrase ""Write Code Here"".

There will be multiple test cases running so the Input and Output should match exactly as provided.

The base Output variable result is set to a default value of 0(Zero) which can be modified. Additionally, you can add or remove these output variables.

Input Format

The first line of input consists of an integer T denoting the number of test cases.

The first line of each test case contains three integers N, P, Q denoting the length of the permutation, the required number of maximas, and the required number of minimas respectively.

Sample Input

3 --> denotes T

4 1 1 --> denotes N,P,Q

6 1 3 --> denotes N,P,Q

6 1 0 --> denotes N,P,Q

Constraints

1≤ T ≤ 1000

1≤ N ≤ 1000

1≤ P, Q ≤ N

Output Format

For each test case, the output contains ""Yes"" if any such permutation is possible, else ""No"".

Print each output on a new line.

Sample Output

Yes

No

Yes

Explanation

In the first test case, the permutations are [1,2,3,4] which can be arranged in the following way [1,3,2,4], where a[1] = 3 will act as local maxima and a[2] =2 will act as local minima.

Hence the answer is ""Yes"".

In the second test case, the permutation is [1,2,3,4,5,6] for which there exists no such arrangement where it can have 1 maxima and 3 minima.

Hence the answer is ""No"".

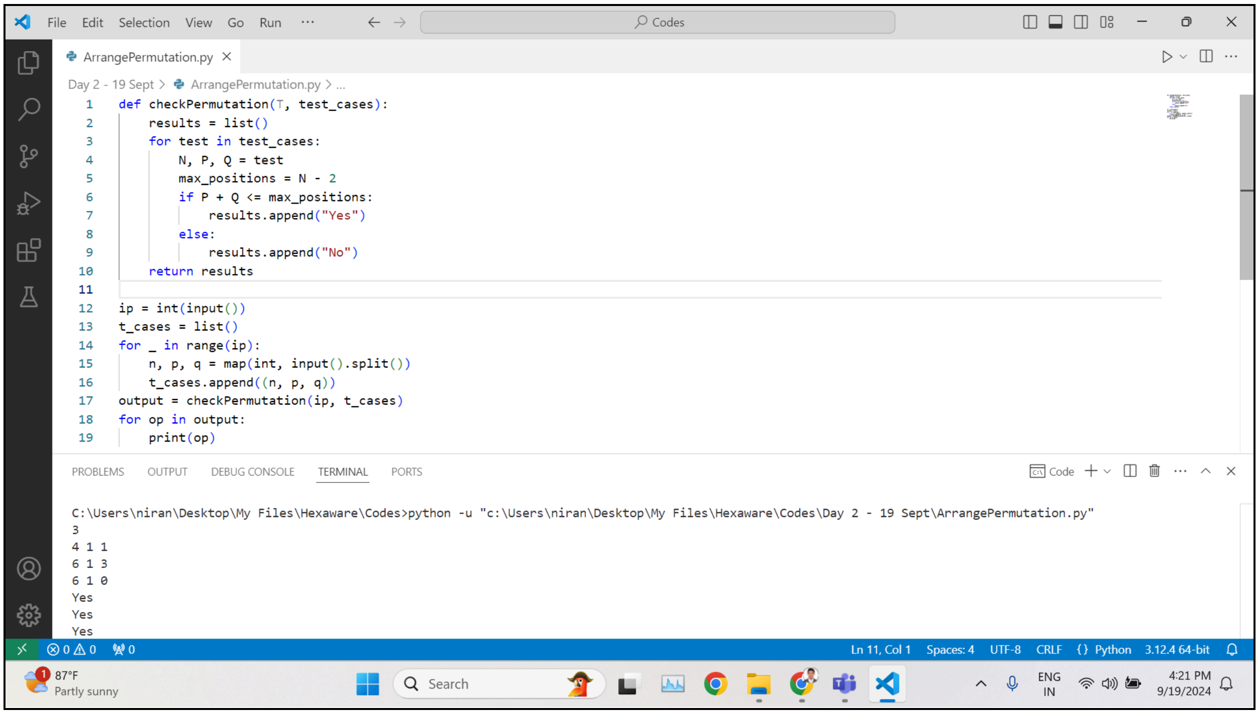
In the third test case, the permutation is [1,2,3,4,5,6] which can be arranged in the following way [1,2,3,4,6,5], where a[4] = 6 will act as local maxima.

Hence the answer is ""Yes""."

Program Code:

```python
def checkPermutation(T, test_cases):
    results = list()
    for test in test_cases:
        N, P, Q = test
        max_positions = N - 2
        if P + Q <= max_positions:
            results.append("Yes")
        else:
            results.append("No")
    return results


ip = int(input())
t_cases = list()
for _ in range(ip):
    n, p, q = map(int, input().split())
    t_cases.append((n, p, q))
output = checkPermutation(ip, t_cases)
for op in output:
    print(op)
```

Output:


File Edit Selection View Go Run ···
Codes
ArrangePermutation.py
Day 2 - 19 Sept > ArrangePermutation.py > ...
def checkPermutation(T, test_cases):
    results = list()
    for test in test_cases:
        N, P, Q = test
        max_positions = N - 2
        if P + Q <= max_positions:
            results.append("Yes")
        else:
            results.append("No")
    return results

ip = int(input())
t_cases = list()
for _ in range(ip):
    n, p, q = map(int, input().split())
    t_cases.append((n, p, q))
output = checkPermutation(ip, t_cases)
for op in output:
    print(op)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
C:\Users\niran\Desktop\My Files\Hexaware\Codes>python -u "c:\Users\niran\Desktop\My Files\Hexaware\Codes\Day 2 - 19 Sept\ArrangePermutation.py"
3
4 1 1
6 1 3
6 1 0
Yes
Yes
Yes
0 0 0
Ln 11, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 64-bit
87°F
Partly sunny
Search
ENG
IN
4:21 PM
9/19/2024