

Task - 12

Elementary data structure algorithm-based Questions, and finding complexity.

file shino and pairs

Given a permutation of numbers from 1 to N. Among all the subarrays, find the number of unique pairs such that and a is maximum and b is second maximum in that subarray.

Algorithm

1. Read the input values of N and the permutation array A.
2. Initialize a variable Count to 0, to keep track the number of valid pairs.
3. Loop over all possible subarrays of A, with nested loops i and j:
 - a. Find the maximum elements maxval and its index maxidx in the subarray [i,j].
 - b. Find the Second maximum element second max in the subarray [i,j] using a separate loop or a priority queue.
 - c. If maxval is unique in the subarray and second max is also present, increment count.
4. Print the final value of Count.

```
#include <stdio.h>
int main() {
    int n, i, j, max, second_max, count = 0; scanf("%d", &n);
    int arr[n];
    for (i = 0; i < n; i++) { scanf("%d", &arr[i]);
    }
    for (i = 0; i < n; i++) { max = arr[i]; second_max = -1;
        for (j = i + 1; j < n; j++) { if (arr[j] > max) {
            second_max = max; max = arr[j];
        }
        else if (arr[j] > second_max) { second_max = arr[j];
        }
        if (second_max != -1 && arr[i] == second_max) { count++;
            break;
        }
    }
    printf("%d", count); return 0;
}
```

Task - 18 (B)

UNSORT
divide the given array of size equal to the sum of
natural numbers into subarrays of size
 $\frac{n}{x}$ for x^{th} subarray.
algorithm

calculate the sum of first n natural numbers.

initialize a variable sum to 0.

loop through the range from 1 to $n+1$.

add each number to the sum variable.

create an array of size equal to the sum calculated
in step 1.

loop through the range from 1 to $n+1$,

If x is odd:

get the subarray of size x starting from the index

Sum - x .

Else:

get the subarray of in descending order.

Add the first element of the subarray to a variable
called result.

Increase the sum of the indices of the subarray by 1

4. Return the variable result as the output.

```
#include <stdio.h>
#include <stdlib.h> int main() {
    int T, n, i, j, k, sum, size;
    scanf("%d", &T);
    while (T--) {
        scanf("%d", &n);
        size = (n * (n + 1)) / 2;
        int *arr = (int *) malloc(sizeof(int) * size); for (i = 0; i < size; i++) {
            scanf("%d", &arr[i]);
        }
        sum = 0;
        k = 0;
        for (i = 1; i <= n; i++) { if (i % 2 != 0) {
            for (j = 0; j < i; j++) { int max = j;
                for (int l = j + 1; l < i; l++) {
                    if (arr[l + k] > arr[max + k]) { max = l;
                }
            }
            int temp = arr[j + k]; arr[j + k] = arr[max + k]; arr[max + k] = temp;
        }
        sum += arr[k]; k++;
    }
    else {
        for (j = 0; j < i; j++) { int min = j;
```

```

for (int l = j + 1; l < i; l++) {
    if (arr[l + k] < arr[min + k]) { min = l;
}
}

int temp = arr[j + k]; arr[j + k] = arr[min + k]; arr[min + k] = temp;

sum += arr[k]; k += i;

printf("%d\n", sum); free(arr);

return 0;
}

```

VEL TECH - CSE

EX NO.	12
PERFORMANCE (5)	8
RESULT AND ANALYSIS (3)	2
VIVA VOCE (3)	2
RECORD (4)	4
TOTAL (15)	16
SIGN WITH DATE	

Result :-

Thus, the program is verified and executed successfully