# Assignment-1 Solution

**Q1(a)** Pseudo-code

**MergeAndInv (A,i,j,k)**

```
B=A
p=i, q=j+1, r=i, s=0
// (1)
while (r≤k) do
// (2)
    if (p≤j) and (q≤k) and (A[p] ≤A[q]):
        {B[r]=A[p], p=p+1, r=r+1, s=s+(q-j-1), continue}
    if (p≤j) and (q≤k) and (A[p] > A[q]):
        {B[r]=A[q], q=q+1, r=r+1, continue}
    if (p≤j) and (q>k):
        {B[r]=A[p], p=p+1, r=r+1, s=s+(k-j), continue}
    if (p>j) and (q≤k):
        {B[r]=A[q], q=q+1, r=r+1, continue}
// (3)
for r =1 to k do
    A[r]=B[r]
return s
```

**SortAndInv(A,i,j)**

**if** (i<j)

k= i + (j-i)//2
x=SortAndInv(A,i,k)
y=SortAndInv(A,k+1,j)
z=MergeAndInv(A, i, k, j)
**return** (x+y+z)

**else return** 0

**Inversions(A)**

B=A
**return** SortAndInv(B, 1, len(A))

**Q1(b)** Correctness.

**Specification** of MergeAndInv (A,i,j,k)

**Input:** sorted segments A[i,j] and A[j+1,k]

**output:** Returns $\sum_{t=i}^{j} Less_{i,j,k}(A[t])$

where $Less_{i,j,k}(x)$ is the no. of elements in $A[j+1,k]$ strictly less than $x$.

That is, $Less_{i,j,k}(x) = |\{y \mid y \in A[j+1,k], y < x\}|$

Also merges segments $A[i,j]$ and $A[j+1,k]$ into sorted $A[i,k]$

**Invariant for while loop has two parts.**

**(i)** $s = \sum_{t=i}^{p-1} Less_{i,j,k}(A[t])$

**(ii)** $p \leq j \rightarrow A[j+1], ..., A[q-1] < A[p]$

**Initialization:** (i) and (ii) hold at (1) trivially as ranges $[i, p-1]$ and $[j+1, q-1]$ are empty.

**Maintenance:** We show that if $(i), (ii)$ and $r \leq k$ hold at (2) then $(i), (ii)$ hold at (3).

Let $p_0, q_0, s_0$ be the value of $p, q$ at (2). We consider each of the four cases in the while loop.

**1.**

**(i)** $A[j+1], ..., A[q_0 - 1] < A[p_0]$, by (ii) at (2).
$A[p_0] \leq A[q_0]$ and $A[j+1,k]$ is sorted
$\Rightarrow A[p_0] \leq A[u]$ for all $u > q_0 - 1$ in $[j+1,k]$

$\Rightarrow Less(A[p_0]) = q - 1 - (j+1) + 1 = q - j - 1$

$\Rightarrow s_0 + (q - j - 1) = \sum_{t=i}^{p_0} Less(A[t])$ [using (i) at (2)]

At (3), $s = s_0 + (q - j - 1)$ and $p = p_0 + 1$

$\Rightarrow$ (i) holds at (3)

**(ii)** If $p_0 + 1 \leq j$, then $A[p_0 + 1] \geq A[p_0]$ (because $A[i, j]$ is sorted)

$\Rightarrow A[j+1], ..., A[q-1] < A[p_0 + 1]$

$\Rightarrow$ (ii) holds at (3)

**2.**

**(i)** $p, s$ are unchanged, so $(i)$ continues to hold at (3).

**(ii)** $A[q_0] < A[p_0] \Rightarrow A[j+1], ..., A[q_0] < A[p_0]$ (using (ii) at (2))

At (3), $q = q_0 + 1$, (ii) also holds at (3)

**3.**

**(i)**

By (ii) at (2), $A[j+1], ..., A[q_k] < A[p_0]$

$\Rightarrow Less(A[p_0]) = k - (j+1) + 1 = k - j$

$\Rightarrow s_0 + (k - j) = \sum_{t=i}^{p_0} Less(A[t])$

At (3), $s = s_0 + (k - j)$ and $p = p_0 + 1$,

$\Rightarrow$ $(i)$ holds at (3).

**(ii)** For (ii), the reasoning is the same as in case 1.

**4.**

**(i)** As $p$ is unchanged, $(i)$ holds at (3)

**(ii)** As $p > j$, $(ii)$ holds at (3) trivially.

**Correctness** (of MergeAndInv (A,i,j,k))

At the end of while loop, we have $(i), (ii)$ and $r > k$

$r > k \rightarrow p = j + 1$ (proved in the correctness of merge sort)

By $(i)$, $s = \sum_{t=i}^{j} Less_{i,j,k}(A[t])$.

As the program MergeAndInv (A,i,j,k) returns this $s$,

the Specification about returned value follows.

Sorting part of the specification has already been proved in mergesort.

**Specification** of SortAndInv(A,i,j)

**Input:** Arbitrary array segment A[i,j]

**output:** Returns numbers of inversions within A[i,j].

Also sorts $A[i,j]$

**Correctness:**

The claim about sorting was proved in lecture.

We prove the claim about inversion by induction on $j-i+1$.

**Base-Case:** $j-i+1 \leq 1$

$\Rightarrow j \leq i$ and the function returns 0.

(As the array segment is empty or has just one element, number of inversions is indeed 0)

**Induction-Step:**

By induction hypothesis,

SortAndInv(A,i,k)=number of inversions within $A[i,k]$.

SortAndInv(A,k+1,j)=number of inversions within $A[k+1,j]$.

By specification of MergeAndInv,

MergeAndInv (A,i,k,j)=$\sum_{t=i}^{k} Less_{i,k,j}(A[t])$

The claim follows by observation that the number of inversions within $A[i,j]$

= number of inversions within $A[i,k]$

+ number of inversions within $A[k+1,j]$

$$+\sum_{t=i}^{k} Less_{i,k,j}(A[t]).$$

(The observation can be proved by considering number of inversions for each element in $A[i,j]$)

**Specification** of Inversions(A)

Returns the number of inversions in A. Does not modify A. Correctness immediately follows from the specification of SortAndInv.