where row = a[j].col, col = a[j].row, and value = a[j].value. Furthermore, these entries occupy the first currentb positions of b[] and are arranged such that: 1) The row fields in b[i] to b[currentb-1] are in ascending order
2) For entries with the same row, the col fields are in ascending order.

5. **Initialization:** Just before the start of the 1st iteration of the outer loop, $i=0$. The set of columns 0 to $c-1=-1$ doesn't make sense and is empty. A universal quantifier over an empty set is vacuously true. ∴ Loop invariant trivially holds.

Maintenance: Let, the loop invariant hold just before the start of the iteration when $i=x$, $0<x<a[0].col$. By the correctness of the inner loop, all non-zero matrix elements of a[] having column x have been correctly transposed in b[] as $\langle a[j].col=x, a[j].row, a[j].value \rangle$ in the index currentb (Lines 11-17). We note that the value of currentb increases by 1 (line 16) with each entry being inserted in b[]. From the correctness of the inner loop we have also seen that for all non-zero elements in b having row x (i.e. column x in a), the elements are arranged in b[] such that their respective column indices are in ascending order.

• It's easy to see that x is the largest indexed column from 0 to x of a[]. Since, we now know that elements of column x has been correctly transposed in b[], by using the loop invariant and correctness of inner loop, we can say: the transpose of all non-zero elements in columns 0 to x of matrix a[] have been correctly placed in b[] as triples $\langle row, col, value \rangle$, where row = a[j].col, col = a[j].row and value = a[j].value. These entries occupy the first currentb positions of b[] and are arranged such that: 1) The row fields from b[1] to b[currentb-1] are in ascending order. 2) For entries with the same row, the col fields are in