the first non-zero entry of row 0 in the transposed matrix should be placed. ∴ Loop invariant holds just before i=1.

Maintenance: Let the loop invariant hold just before the start of the iteration when i=x, where 1<x<numCols. Therefore, for all indices K where 0≤K<x, startingPos[K] correctly stores the index in array b[] where the first non-zero entry of row K in the transposed matrix should be placed.

- By the definition of the transpose matrix, row x of matrix b[] is actually the column x of matrix a[]. By the correctness of the loop in lines 10-11, rowTerms[x-1] stores the no. of non-zero elements of column (x-1) of a[], hence it is also the no. of non-zero elements of row (x-1) of b[] (We intend to create matrix b[] as the transpose of matrix a[]).
- From the above justification, it's easy to see why assigning the value of startingPos[x-1]+rowTerms[x-1] to the storage location startingPos[x] is correct. (Justification of line 14)
- The loop variable i updates to (x+1) and it's easy to see that the loop invariant still holds.

Termination: The loop terminates when i=numCols. By the loop invariant, for all indices K where 0≤K<numCols, startingPos[K] correctly stores the index in array b[] where the first non-zero entry of row K in the transposed matrix should be placed.

- By the defn. of transpose, we know that column K of a[] is row K of b[]. ∴ By criterion of a[], startingPos[K] correctly stores the index in array b[] of the element having column K in array a[] and minimum indexed row of all other non-zero elements of a[] having column K. Since the index K is in the range 0 to numCols-1, startingPos[] has taken into account all the columns of a[] □