```
void transpose() {
    int num_rows = row_pointers_size - 1;
    int num_cols = -1;
    for (int k = 0; k < num_nonzeros; ++k) {
        num_cols = max(num_cols, tuple_table[k][1] + 1);
    }

    int* col_counts = new int[num_cols];
    memset(col_counts, 0, sizeof(int) * num_cols);
    for (int k = 0; k < num_nonzeros; ++k) {
        col_counts[tuple_table[k][1]]++;
    }

    int* transposed_row_pointers = new int[num_cols + 1];
    memset(transposed_row_pointers, 0, sizeof(int) * (num_cols + 1));
    for (int j = 1; j <= num_cols; ++j) {
        transposed_row_pointers[j] = transposed_row_pointers[j-1] + col_counts[j-1];
    }

    int** transposed_tuple_table = new int*[num_nonzeros];
    for (int k = 0; k < num_nonzeros; ++k) {
        int row = tuple_table[k][0];
        int col = tuple_table[k][1];
        double value = tuple_table[k][2];

        int transposed_row = transposed_row_pointers[col];
        transposed_tuple_table[transposed_row] = new int[3]{col, row, value};
        transposed_row_pointers[col]++;
    }

    for (int j = num_cols; j > 0; --j) {
        transposed_row_pointers[j] = transposed_row_pointers[j-1];
    }
    transposed_row_pointers[0] = 0;

    for (int k = 0; k < num_nonzeros; ++k) {
        delete[] tuple_table[k];
    }
    delete[] tuple_table;

    row_pointers = transposed_row_pointers;
    tuple_table = transposed_tuple_table;
    row_pointers_size = num_cols + 1;
    num_nonzeros = tuple_table_size;
}
```

```
int row_pointers[rowsize + 1];
int* tuple_table[nonzerosize];
int row_pointers_size;
int tuple_table_size;
int num_nonzeros;
```