MP7

1. Consider a sparse matrix with m rows, n columns, k non-zero integers elements, and the maximum number of non-zero elements in a single row is p. How many bytes would be needed to represent the matrix in COO format? Assume sizeof(int) = 4 bytes.

a. 8k+4m+4 bytes
b. 8k+4n bytes
c. 12k bytes
d. 8mp bytes
e. 12mp bytes

Solution: (c) 12k bytes. Each non-zero element has a row and a col index. Thus, 3k integers, hence, 12k bytes. **1 point**

2. Consider a sparse matrix with m rows, n columns, k non-zero integers elements, and the maximum number of non-zero elements in a single row is p. How many bytes would be needed to represent the matrix in ELL format? Assume sizeof(int) = 4 bytes.

a. 8k+4m+4 bytes
b. 8k+4n bytes
c. 12k bytes
d. 8mp bytes
e. 12mp bytes

Solution: (d) 8mp bytes 4mp bytes for data and 4mp bytes for col_index. **1 point**

3. Given the following sparse matrix, [[1, 0, 3, 2, 0], [6, 8, 7, 9, 10], [13, 4, 0, 0, 0], [0, 0, 5, 0, 0], [12, 11, 0, 15, 14]], how many bytes would be needed to represent the matrix in JDS-transposed format? Assume sizeof(int) = 4 bytes.

a. 40 bytes
b. 160 bytes
c. 45 bytes
d. 180 bytes
e. 200 bytes

Solution: (d) 180 bytes. matData and matCols both have 15 elements. matColStart, matRowPerm and matRows all have 5 elements. So the total number of bytes is (15*2+5*3)*4=180 bytes. **1 point**

NOTE: Since one can discard the matRows and still perform arithmetic operations by slightly modifying the code given in lecture 20, the total number of bytes can also be (15*2+5*2)*4=160 bytes. Thus we also accept **(b) 160 bytes**.

4. Convert the following JDS-Transposed representation to a possible ELL representation i.e. the data and the col_index. JDS-Transposed representation is as follows (the variables represent the same data-structures as the MP code assignment):

```
matData[] = {1, 5, 4, 2, 6, 3};
matCols[] =  {0, 0, 2, 2, 2, 3};
matColStart[] =  {0, 3, 5, 6};
matRowPerm[] =  {0, 3, 2, 1};
matRows[] =  {3, 2, 1, 0};
```

Solution:
For data[], we accept these answers:
{1, *, 4, 5, 2, *, *, 6, 3, *, *, *};
{1, 0, 4, 5, 2, 0, 0, 6, 3, 0, 0, 0};

For col_index[], we accept these answers:
{0, *, 2, 0, 2, *, *, 2, 3, *, *, *};
{0, 0, 2, 0, 2, 0, 0, 2, 3, 0, 0, 0};
{0, 0, 2, 0, 2, 1, 3, 2, 3, 2, 0, 3};

Original matrix(**for visualization purpose only**) =
1 0 2 3
0 0 0 0
0 0 4 0
5 0 6 0
1 point for correct data and 1 point for correct col_index

5. Given the following sparse matrix, [[7, 0, 3, 0, 0], [0, 2, 5, 1, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 7]], write down its COO representations in three arrays (Hint: data, col_index, row_index):

Solution:
data[6] = {7, 3, 2, 5, 1, 7};
col_index[6] = {0, 2, 1, 2, 3, 4};
row_index[6] = {0, 0, 1, 1, 1, 3};
1 point for getting all three arrays correct. No partial credit

6. Translate the COO representations in the previous question into JDS format. Write down the representations in four arrays (Hint: data, col_index, jds_row_ptr, jds_row_perm):

Solution:
data[6] = {2, 5, 1, 7, 3, 7};
col_index[6] = {1, 2, 3, 0, 2, 4};
jds_row_ptr[5] = {0, 3, 5, 6, 6};
jds_row_perm[5] = {1, 0, 3, 2};

1 point for getting half of the arrays correct(any combinations) and 2 points for getting all results correct

7. Given the following CSR Row Pointers, [0, 3, 5, 6, 6, 8, 12, 12, 12, 19], how many rows, how many non-zero elements, how many empty rows are there in the matrix? Given the information, what's the minimum possible elements this matrix can contain? (Hint: what's the minimum size can we set given the row_ptr array)

Solution:
9 rows (length - 1)
19 non-zeros elements (given the largest number)
3 empty rows (count number of consecutive same integers)
minimum 9 * 7 = **63** (last row 12 - 19 has 7 elements and thus we need to have at least 7 cols)

1 point for getting # of rows, # of non-zero elements and # of empty rows correct and 1 point for getting minimum # of elements in the matrix correct