National University of Singapore
School of Computing
CS1010X: Programming Methodology
Semester II, 2024/2025

Recitation 7
Multiple Representations

Problems

1. **Dense Matrix Representation**. A matrix can be represented in Python by a list of lists (nested lists). For example, `m = [[ 1, 2, 3], [4, 5, 6], [7, 8, 9]]` represents the following 3×3 matrix:

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

You are given the following implementation for `make_matrix(seq)`, which takes in a sequence, i.e. either a tuple or a list, and creates the matrix object.

```python
def make_matrix(seq):
    mat = []
    for row in seq:
        mat.append(list(row))
    return mat
```

(a) Suppose `seq` were a list of lists. Would the following implementation of `make_matrix(seq)` work? Explain.

```python
def make_matrix(seq):
    return seq
```

(b) Implement the following supporting functions:

   i. `rows(m)`: returns the number of rows for matrix object `m`.

   ii. `cols(m)`: returns the number of columns for matrix object `m`.

iii. `get(m,i,j)`: returns the element (i,j) for matrix object `m`.

iv. `set(mat,i,j,val)`: sets the element (i,j) for matrix object `m` to value `val`.

v. `transpose(m)`: transposes matrix object `m`. Basically, this converts a $m \times n$ matrix into a $n \times m$ matrix.

vi. `print_matrix(mat)`: prints the contents of matrix object `m` in a human readable form.

2. **Sparse Matrix Representation**. Now suppose that implementation of
   `make_matrix(seq)` is as follows:

```python
def make_matrix(seq):
    data = []
    for i in range(len(seq)):
        for j in range(len(seq[0])):
            if seq[i][j] != 0:
                data.append([i,j,seq[i][j]])
    return [len(seq),len(seq[0]),data]
```

(a) Implement the list of associated functions listed in Part 1(ii) above.

   i. `rows(m)`

   ii. `cols(m)`

   iii. `get(m,x,y)`

   iv. `set(mat,x,y,val)`

     v. `transpose(m)`

     vi. `print_matrix(mat)`

(b) Which is the better implementation for the matrix object? Explain.