CS 331 Midterm Exam Worksheet

<table>
<thead>
<tr>
<th>WP1</th>
<th>(/8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP2</td>
<td>(/8)</td>
</tr>
<tr>
<td>WP3</td>
<td>(/8)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>(/24)</td>
</tr>
</tbody>
</table>
Python data structures

WP1

```python
def zip(*seqs):
    its = [iter(l) for l in seqs]
    zl = []
    while True:
        try:
            zl.append(tuple(next(it) for it in its))
        except:
            return zl

OR

def zip(*seqs):
    min_length = len(seqs[0])
    for i in range(1, len(seqs)):
        if len(seqs[i]) < min_length:
            min_length = len(seqs[i])
    ziplist = []
    for i in range(min_length):
        ziplist.append(tuple(l[i] for l in seqs))
    return ziplist
```
Mystery sort

WP2(a)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>0</th>
<th>7</th>
<th>5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WP2(b)

O(N^2)

WP2(c)

Prefer insertion sort, as it does better than this mystery sort function on partially ordered lists.
Array-backed list

WP3

class ArrayList:
    def __init__(self):
        self.data = []

    def move(self, idx_src, idx_dst):
        x = self.data[idx_src]
        if idx_dst < idx_src:
            for i in range(idx_src, idx_dst, -1):
                self.data[i] = self.data[i-1]
        else:
            for i in range(idx_src, idx_dst):
                self.data[i] = self.data[i+1]
        self.data[idx_dst] = x