3) see Chapter3-ShortAnswer3.py for the implementation.

Here is the running time for each of four requested operations:

<table>
<thead>
<tr>
<th>operations</th>
<th>(a) the objects are stored in a Python list in the order that they are added</th>
<th>(b) the objects are stored in a Python list in order by name</th>
<th>(c) the objects are stored in a Python dictionary indexed by name</th>
</tr>
</thead>
</table>
| add(person)    | $\Theta(1)$  
we discussed that addition of a value to the Python's list is on average $\Theta(1)$ operation (even though sometimes we have to allocate new space and copy elements there)  
|                | $\Theta(\log n)$  
since the elements are ordered, we need to find a position to insert the new record, search can be done with $\log n$ time (recall binary search on sorted arrays). | $\Theta(1)$  
almost all basic operations on dictionaries are $\Theta(1)$, since hash tables with hashing function are used. |
| remove(name)   | $\Theta(n)$  
operations of insertion and deletion are $\Theta(n)$ for Python's lists + we need to find a record with that name, and since elements are no sorted it is also $\Theta(n)$, which results is $\Theta(n)$ | $\Theta(n)$  
first we will need to locate the element with the given name ( $\Theta(\log n)$ operation), then we will need to delete is ($\Theta(n)$ operation on Python's lists), hence the result is $\Theta(n)$ | $\Theta(1)$  
using hashing function the record will be accessed in constant time, and deleted |
| lookup(name)   | $\Theta(n)$  
search operation on an unsorted array/list is $\Theta(n)$ | $\Theta(\log n)$  
search in a sorted array is $\Theta(\log n)$ operation | $\Theta(1)$  
using hashing function the record will be accessed in constant time |
| list_all       | $\Theta(n \log n)$  
sorting unsorted list takes at least (based on what we learned so far) $\Theta(n \log n)$ time | $\Theta(1)$  
the elements are already sorted | $\Theta(n \log n)$  
copying elements into a list is $\Theta(n)$ + sorting is at least $\Theta(n \log n)$, hence overall we have $\Theta(n \log n)$ |