Code Exercises Report

1.Implement the methods specified by the Grid interface using this data structure. Why is this a more time-efficient implementation than BoundedGrid?

Answer: According to the description, there is a program requires a very large bounded grid that contains very few objects and that the program frequently calls the getOccupiedLocations method. The time complexity of the getOccupiedLocations method in BoundedGrid implementation is O(r * c), where r is the row number of the grid, c is the col number of the grid. But implement this method using this data structure, the time complexity of this method will decrease to O(r + n), where r is the row number of the grid and n is the number of items in the grid. Because the grid is sparse, r + n is far less than r * c, so it's more time-efficient.

2.. How could you use the UnboundedGrid class to accomplish this task? Which methods of UnboundedGrid could be used without change?

Answer: Most mothods in UnboundedGrid class are useful to accomplish this task. The getOccupiedLocations, get, put, remove methods can be used without change. So we only need implement the isValid, getNumRows, getNumCols method.

Methods	SparseGridNode version	LinkedList<OccupantInCol> version	HashMap version	TreeMap version
getNeighbors	O(c)	O(c)	O(1)	O(log n)
getEmptyAdjacentLocations	O(c)	O(c)	O(1)	O(log n)
getOccupiedAdjacentLocations	O(c)	O(c)	O(1)	O(log n)
getOccupiedLocations	O(r + n)	O(r + n)	O(n)	O(n)
get	O(c)	O(c)	O(1)	O(log n)
put	O(c)	O(c)	O(1)	O(log n)
remove	O(c)	O(c)	O(1)	O(log n)

3.Implement the methods specified by the Grid interface using this data structure. What is the Big-Oh efficiency of the get method? What is the

efficiency of the put method when the row and column index values are within the current array bounds? What is the efficiency when the array needs to be resized?

Answer: The Big-Oh efficiency of the get method is O(1). The efficiency of the put method is O(1) when the row and column index values are within the current array bounds. The efficiency is O(len * len) when the array needs to be resized, where len is the number of rows and cols in the current bound.