Part5

Do you Know?

Set10

1. Where is the isValid method specified? Which classes provide an implementation of this method?
   这个特别的方法在grid中，UnboundedGrid类和BoundedGrid类提供了对这个方法的implementation，如下：

```
//Grid.java
boolean isValid(Location loc);

//UnboundedGrid.java
public boolean isValid(Location loc)
{
    return true;
}

//BoundedGrid.java
public boolean isValid(Location loc)
{
    return true;
}
```

2. Which AbstractGrid methods call the isValid method? Why don't the other methods need to call it?

   AbstractGrid中的getValidAdjacentLocations调用了isValid方法；

   因为其他的方法都直接或者间接的使用了此方法，所以不需要重复调用isValid（）。

```
 //AbstractGrid.java
if (isValid(neighborLoc))
    locs.add(neighborLoc);
```

3. Which methods of the Grid interface are called in the getNeighbors method? Which classes provide implementations of these methods?

方法：getOccpiedAdjacentLocations(),get()

AbstractGrid类提供了对getOccpiedAdjacentLocations()的implements

BoundedGrid 类和UnboundedGrid类提供了对get()的implements

```
//AbstractGrid.java
for (Location neighborLoc : getOccupiedAdjacentLocations(loc))
    neighbors.add(get(neighborLoc));

//bstractGrid.java
public ArrayList<Location> getOccupiedAdjacentLocations(Location loc)
{
    ArrayList<Location> locs = new ArrayList<Location>();
    for (Location neighborLoc : getValidAdjacentLocations(loc))
    {
        if (get(neighborLoc) != null)
            locs.add(neighborLoc);
    }
    return locs;
}

//BoundedGrid.java

public E get(Location loc)

//UnboundedGrid.java
public E get(Location loc)
```

4. Why must the get method, which returns an object of type E, be used in the getEmptyAdjacentLocations method when this method returns locations, not objects of type E?

   由于get方法返回存储在给定位置的网格中的对象时，会返回此对象的引用，如果没有对象，则返回null，故需要返回E类型的对象。

   getEmptyAdjacent方法通过使用get方法来判断返回的结果是否为null，也正因如此我们可以得到正确的空区。

```
//AbstractGrid.java
public ArrayList<Location> getEmptyAdjacentLocations(Location loc)
{
    ArrayList<Location> locs = new ArrayList<Location>();
    for (Location neighborLoc : getValidAdjacentLocations(loc))
    {
```

```
        if (get(neighborLoc) == null)
            locs.add(neighborLoc);
    }
    return locs;
}
```

5. What would be the effect of replacing the constant Location.HALF_RIGHT with Location.RIGHT in the two places where it occurs in the getValidAdjacentLocations method?

   会使得有效相邻位置的最大数量从 8 个减少到 4 个。只有东，西，南，北。

```
//AbstractGrid.java
int d = Location.NORTH;
for (int i = 0; i < Location.FULL_CIRCLE / Location.HALF_RIGHT; i++)
{
    Location neighborLoc = loc.getAdjacentLocation(d);
    if (isValid(neighborLoc))
        locs.add(neighborLoc);
    d = d + Location.HALF_RIGHT;
}
```

Set11

1. What ensures that a grid has at least one valid location?

   BoundedGrid的构造函数决定的：

```
//BoundedGrid.java
public BoundedGrid(int rows, int cols)
{
    if (rows <= 0)
        throw new IllegalArgumentException("rows <= 0");
    if (cols <= 0)
        throw new IllegalArgumentException("cols <= 0");
    occupantArray = new Object[rows][cols];
}
```

2. How is the number of columns in the grid determined by the getNumCols method? What assumption about the grid makes this possible?

   是由getNumCols()的return值决定的，BoundedGrid至少有一行一列。

```java
//BoundedGrid.java
public int getNumCols()
{
    return occupantArray[0].length;
}
```

3. What are the requirements for a Location to be valid in a BoundedGrid? In the next four questions, let r = number of rows, c = number of columns, and n = number of occupied locations.

   0≤the row of the location < the number of rows

   0≤the col of the location < the number of colums

```java
//BoundedGrid.java
public boolean isValid(Location loc)
{
    return 0 <= loc.getRow() && loc.getRow() < getNumRows()
        && 0 <= loc.getCol() && loc.getCol() < getNumCols();
}
```

4. What type is returned by the getOccupiedLocations method? What is the time complexity (Big-Oh) for this method?

   返回类型：ArrayList<Location>

   时间复杂度：O（r*c）

```java
//BoundedGrid.java
public ArrayList<Location> getOccupiedLocations()
{
    ArrayList<Location> theLocations = new ArrayList<Location>();

    // Look at all grid locations.
    for (int r = 0; r < getNumRows(); r++)
    {
        for (int c = 0; c < getNumCols(); c++)
        {
            // If there's an object at this location, put it in the array.
            Location loc = new Location(r, c);
            if (get(loc) != null)
                theLocations.add(loc);
        }
    }
```

```
17
18      return theLocations;
19  }
```

5. What type is returned by the get method? What parameter is needed? What is the time complexity (Big-Oh) for this method?

   返回类型：E；

   参数类型：Location

   时间复杂度：O（1）

```
1  //BoundedGrid.java
2  public E get(Location loc)
3  {
4      if (!isValid(loc))
5          throw new IllegalArgumentException("Location " + loc
6                                             + " is not valid");
7      return (E) occupantArray[loc.getRow()][loc.getCol()]; // unavoidable warn
8  }
```

6. What conditions may cause an exception to be thrown by the put method? What is the time complexity (Big-Oh) for this method?

   两种情况：

   location不是有效的，即isValid(location)==0;

   传进来的object为空

   时间复杂度：O（1）

```
1  //BoundedGrid.java
2  public E put(Location loc, E obj)
3  {
4      if (!isValid(loc))
5          throw new IllegalArgumentException("Location " + loc
6                                             + " is not valid");
7      if (obj == null)
8          throw new NullPointerException("obj == null");
9
10     // Add the object to the grid.
11     E oldOccupant = get(loc);
12     occupantArray[loc.getRow()][loc.getCol()] = obj;
13     return oldOccupant;
```

```
14 }
```

7. What type is returned by the remove method? What happens when an attempt is made to remove an item from an empty location? What is the time complexity (Big-Oh) for this method?

   返回类型：E

   如果这个位置是isValid的，则不会有错误，否则会出现错误。

   时间复杂度：O（1）

8. Based on the answers to questions 4, 5, 6, and 7, would you consider this an efficient implementation? Justify your answer.

   本小组统一这是一种有效的implementation，因为时间复杂度都较低，最大的也不过为O(r*c)。

Set12

1. Which method must the Location class implement so that an instance of HashMap can be used for the map? What would be required of the Location class if a TreeMap were used instead? Does Location satisfy these requirements?

   HashMap：Location类实现了hashCode()方法和equals()方法；

   TreeMap：Location实现了Compare接口，并且需要实现compareTo()，同时TreeMap需要key是可以进行比较的。

```java
//Location.java
public class Location implements Comparable

public int hashCode()
{
    return getRow() * 3737 + getCol();
}

public boolean equals(Object other)
{
    if (!(other instanceof Location))
        return false;
    Location otherLoc = (Location) other;
    return getRow() == otherLoc.getRow() && getCol() == otherLoc.getCol();
}
```

2. Why are the checks for null included in the get, put, and remove methods? Why are no such checks included in the corresponding methods for the BoundedGrid?

在UnboundedGrid中，数据存储在哈希映射中，键是位置，值是相应位置中的对象。哈希映射的键不能为空。因此，我们需要检查参数位置是否不为 null。

在BoundedGrid中，数据存储在数组中。我们检查位置是否在方法中有效。因此，如果它为null，则也会引发异常。

3. What is the average time complexity (Big-Oh) for the three methods: get, put, and remove? What would it be if a TreeMap were used instead of a HashMap?

   get,put,remove的平均时间复杂度：O（1）；

   TreeMap的平均时间复杂度：O（logN）；

4. How would the behavior of this class differ, aside from time complexity, if a TreeMap were used instead of a HashMap?

   对于getOccupieLocations方法获取占领位置，当使用哈希映射时，返回的信息大多以不同的顺序排列，因为HashMap中的key被放置在哈希表中，然后我们通过key获取它。但是，如果我们使用TreeMap，我们会将数据存储在二叉树中，并以稳定的顺序遍历此树（不排除无序遍历的情况）。

5. Could a map implementation be used for a bounded grid? What advantage, if any, would the two-dimensional array implementation that is used by the BoundedGrid class have over a map implementation?

   map是可以用在bounded grid的；

   采用HashMap：get,put,remove的平均时间复杂度：O（1），getOccupiedLocations()的平均时间复杂度为：O（n）

   采用TreeMap:get,put,remove的平均时间复杂度：O（nlogn）,getOccupiedLocations()的平均时间复杂度为：O（n）

   相较之下，HashMap更省一些时间，但总体的复杂度都不高，都可以接受。

   同时采用2维数组也有优势，如果网格中有多个物体，2维数组只需要存储网格相应位置的对象，而Map需要同时存储位置和对象。