1 Heap

(a) Suppose we have the min heap below, with array representation as shown. Show the heap and array representation after the smallest value is removed, using the procedure described in class.

![Heap Diagram]

Initial:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Your Answer:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 Left-Leaning Red-Black Tree

(a) Draw the 2-3 tree corresponding to the following left-leaning red-black tree.

![2-3 Tree Diagram]

(b) Draw the left-leaning red-black tree after inserting 5. Label red edges red.

![Updated 2-3 Tree Diagram]
3 Hashing

For the following problems, assume that:

- IntList is a list of integers.
- The hash code of an IntList is the sum of the integers in the list.
- IntLists are considered equal only if they have the same size and the same values in the same order.
- FourBucketHashMap uses separate chaining and that new items are added to the back of each bucket.
- FourBucketHashMap always has four buckets and never resizes.

(a) Draw the hash table that is created by the following code. The result of the first put is provided for you.

```
1 FourBucketHashMap<IntList, String> fbhm = new FourBucketHashMap<>();
2 fbhm.put(IntList.of(1, 2), "dog");
3 fbhm.put(IntList.of(3, 1), "bear");
4 fbhm.put(IntList.of(9), "rat");
5 fbhm.put(IntList.of(3, 3, 2), "tiger");
```

```
0  ←→
|
1  ←→
|
2  ←→
| 1, 2, "dog"  |
3  ←→
```

(b) Consider the following code:

```
1 FourBucketHashMap<IntList, String> fbhm = new FourBucketHashMap<>();
2 IntList list1 = IntList.of(1, 2);
3 fbhm.put(list1, "dog");
4 \ Part i
5 list1.add(3);
6 \ Part ii
```

i) At Part i (line 4), what will be returned from the following statement?

```
fbhm.get(IntList.of(1, 2));  ○ "dog" ○ [1, 2] ○ null
```

ii) At Part ii (line 6), what will be returned from the following statements?

```
fbhm.get(IntList.of(1, 2));  ○ "dog" ○ [1, 2] ○ null
fbhm.get(IntList.of(1, 2, 3));  ○ "dog" ○ [1, 2, 3] ○ null
```

iii) Is there a problem with the code? If so, explain below.