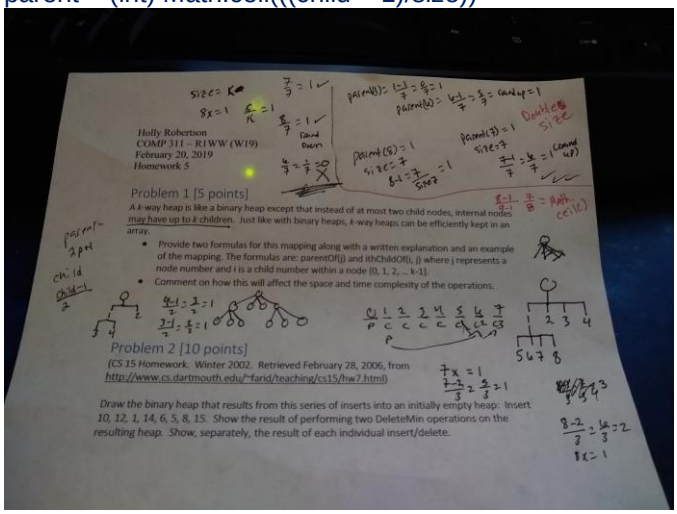
Holly Robertson
COMP 311 – R1WW (W19)
February 20, 2019
Homework 5

## Problem 1 [5 points]

A *k*-way heap is like a binary heap except that instead of at most two child nodes, internal nodes may have up to *k* children. Just like with binary heaps, *k*-way heaps can be efficiently kept in an array.

- Provide two formulas for this mapping along with a written explanation and an example of the mapping. The formulas are: parentOf(j) and ithChildOf(i, j) where j represents a node number and i is a child number within a node {0, 1, 2, … k-1}.
- Comment on how this will affect the space and time complexity of the operations.

parentOf()
k = size (double)
parent = (int) Math.ceil(((child – 1)/size))

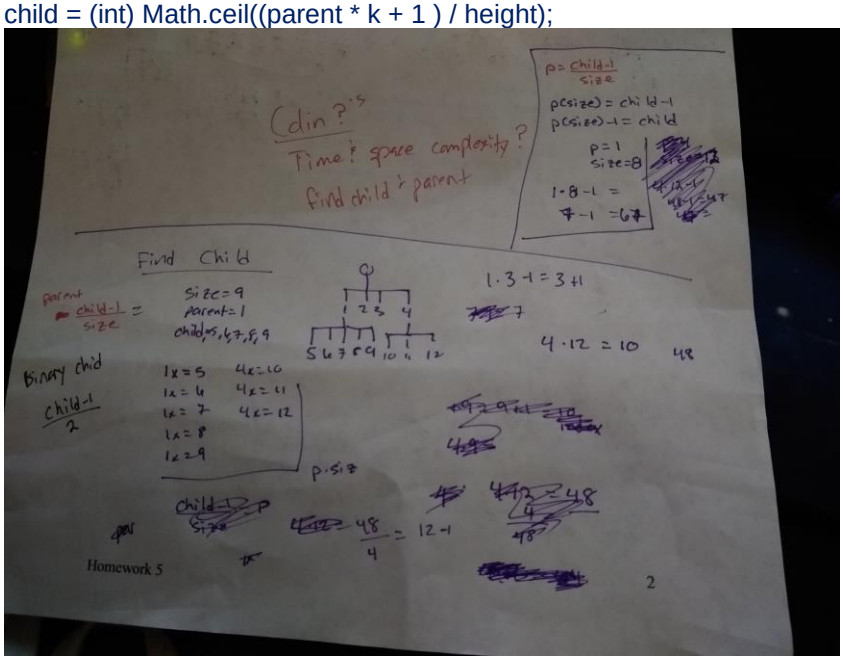



KHeap.java

```java
package hw5;

public class KHeap
{
    public int findParent(double size, int child) {
        return (int) (Math.ceil((child - 1) / (size)));
    }
}
```

```java
package hw5;

import java.util.ArrayList;

public class KHeapTest extends TestCase
{
    KHeap heap;
    ArrayList<Integer> list = new ArrayList<Integer>();
    int child = 0;
    int parent = 0;

    protected void setUp() throws Exception {
        heap = new KHeap();
        for (int i = 0; i < 9; i++) {
            list.add(i);
        }
        child = list.get(list.size() - 1);
        parent = 1;

    }
    protected void tearDown() throws Exception {
        heap = null;
    }

    @Test
    public void testFindParent() {
        assertEquals(1, (heap.findParent(list.size(), child)));
    }
```

ithChildOf()
k = size (double)
child = (int) Math.ceil((parent * k + 1 ) / height);




KHeap.java

```java
package hw5;

public class KHeap
{
    public int findParent(double size, int child) {
        return (int) (Math.ceil((child - 1) / (size)));
    }

    public int findChild(int size, int parent, double height) {
        return (int) Math.ceil((parent * size + 1 ) / height);
    }
}
```

```java
@Test
public void testFindChild() {
    assertEquals(5, (heap.findChild(list.size(), parent, 3 - 1)));
}
```

From what I read about Space Complexity and talking with Colin: “The space complexity of a program (for a given input) is the number of elementary objects that this program needs to store during its execution. This number is computed with respect to the size n of the input data.” [StackOverflow]

So, if we have n elements for the ArrayList and we need to size and child variables to find parentOf(), then S(parentOf()) = n + 2

For ithChildOf() we need size, parent, and height. S(ithChildOf()) = n + 3;

Time complexity of both methods would be (O log n) since the k-way heap is an Array List and retrieval/deletion in an ArrayList is (O log n) where n is the number of items in the array. The max number of nodes in the array/k-way heap would be 2h – 1 and the lowest would be h if h starts at 0.

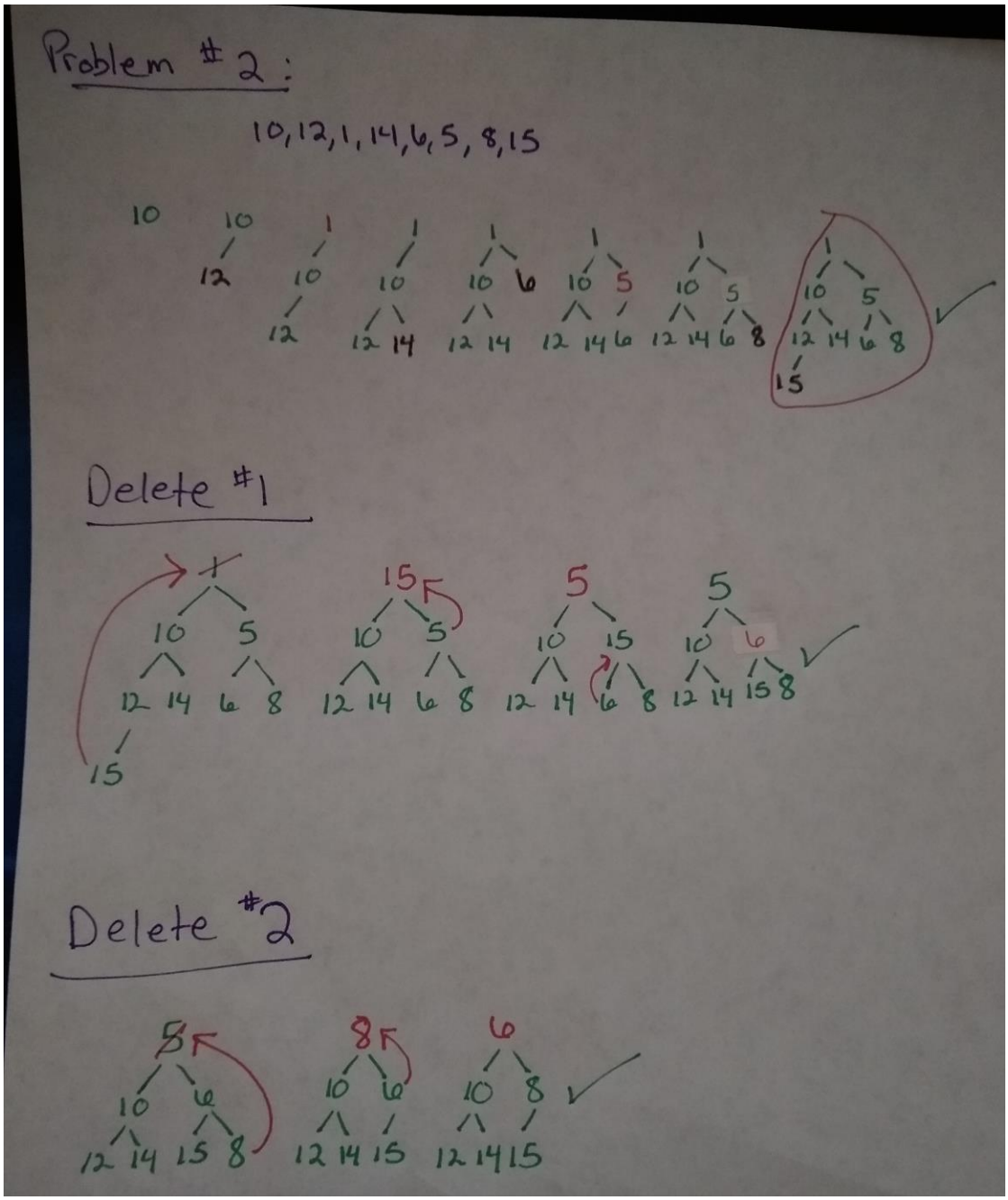
Time Complexity of parentOf() – (O log n)
Space Complexity of parentOf() – n + 2

Time Complexity of ithChildOf() – (O log n)
Space Complexity of ithChildOf() – n + 3

## Problem 2 [10 points]

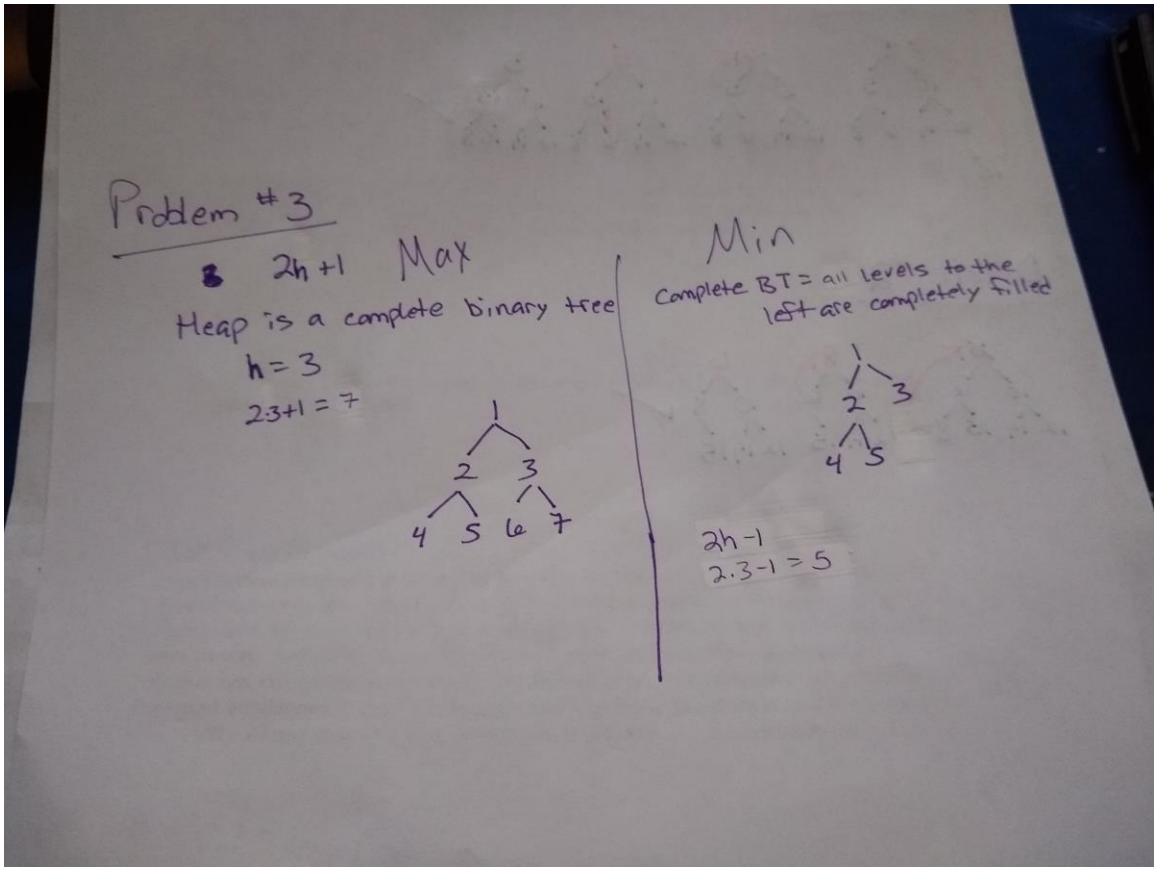
(CS 15 Homework. Winter 2002. Retrieved February 28, 2006, from )

Draw the binary heap that results from this series of inserts into an initially empty heap: Insert 10, 12, 1, 14, 6, 5, 8, 15. Show the result of performing two DeleteMin operations on the resulting heap. Show, separately, the result of each individual insert/delete.



## Problem 3 [5 points]

What are the minimum and maximum numbers of elements in a heap of height *h* assuming that we start counting height at 0?



## Reflection [5 points]

In two to three paragraphs of prose (i.e. sentences, not bullet lists) using APA style citations if needed, summarize and interact with the content that was covered in the class session this week. In your summary, you should highlight the major topics, theories, practices, and knowledge that were covered. Your summary should also interact with the material through personal observations, reflections, and applications to the field of study. In particular, highlight what surprised, enlightened, or otherwise engaged you. Make sure to include at least one thing that you're still confused about. In other words, you should think and write critically not just about what was presented but also what you have learned through the session. Feel free to ask questions in this as well since it will be returned to you with answers.

During the lecture when Tim went over space complexity, I was completely lost. I worked on the algorithms for this week's homework but could not grasp the space complexity – I think I was making it harder than necessary. According to StackOverflow:

1. Time complexity --> CPU usage
2. Space complexity --> RAM usage

If this is right, this makes complete sense. How much space we are using in memory for an algorithm. This will be my question this week.

Another thing I learned is that not to push WebCAT submissions just to test for every small change! I thought we had 90+ attempts before we saw any penalty's, but I was clearly wrong! I need to organize my submissions better for the next lab. Colin gave me some good tips on how to do this. I wasn't utilizing the local checkstyle file on my project – I am now!

My question this week – how would you explain space complexity to a novice?