HW3 Grading Rubric

Written (35):

Problem 1 (7 pts): Weiss 4.6

- +1 for correct base case
    - Base case should be either number of full nodes = 0 or 1
- +2 for correct inductive hypothesis/assumption
- +4 for correct induction step
    - (-2) for recursively removing parent node (doesn't go from k to k+1).
    - (-2) for good attempt at inductive step, but overall incorrect
    - -1 if mentions only one of two cases:
        - two nodes to leaf
        - one node to single-childed node

+1 points total if a correct mathematical proof that doesn't use induction is provided. The question asks for a complete inductive proof.

Problem 2 (7 pts): Weiss 4.9

- Part A (4 pts)
    - -1 for each mistake
    - Only 1 point total if only the final tree is shown and no intermediate steps are shown.
- Part B (3 pts)
    - OK if incorrect due to error in A
    - -3 if incorrect

Problem 3 (7 pts):

- +2 for correctly shown single rotation (Points will not be awarded if only the final tree of the rotation is shown)
- +5 for correct sequence of trees (work must be shown)
    - -1 for a single mistake
    - -2 for 2 mistakes
    - -3 for 3 or more mistakes (including not showing each step)
- No points awarded for plugging numbers into AVL Tree Visualizer

Problem 4 (7 pts):

- +7 points for correct full proof
- +4 points if proof is going in the right direction, but is not fully complete or has ambiguity.
- Adjust +/- partial credit at grader's discretion

Problem 5 (7 pts):

- +2 for correct modification of BinaryNode class (i.e. boolean marker or setting data to null)
    - -1 if the description is vague
- +5 for correct modification of findMin() class
    - -2 if not recursive
    - -2 if the code returns the wrong undeleted value (Assumes the code for findMin handles lazy deletion correctly)
    - -1 if the code returns a deleted value

NOTES:

- Total -2 for not going to right subtree during backtrack
- Total -3 for right logic, but incorrect implementation overall

Programming (65)

- Compiling Issues:
- -2 Simple Fix to Compile (Removing package statements)
- -4 Slightly more complicated issue
- Major Issue: Max ⅔ of total grade based on attempt (student gets 12)

Problem 1 (35)

- +1 point total for correctly named classes
    - No points if any of the classes are named wrong.
    - Leeway on naming ExpressionNodes class (no points off)
- +4 for correctly implemented tester class Problem1
    - Okay if expressions are hardcoded.
    - -1 for command line arguments
    - -3 if any of the functionality that needs to be in the tester is in ExpressionTree.java. (i.e. Instantiating the expression)
    - -1 per method not explicitly demonstrated in the tester class
- +6 for correctly implemented stack-based tree building algorithm
    - -2 pts if algorithm is implemented outside the ExpressionTree constructor
    - -2 pts if ExpressionNode is not a nested class
    - OK if a tree is pushed onto the stack instead a Node object
    - -2 if not stack-based algorithm specified in class
    - -1 for each method signature that the student changes (return type or parameter)
- +6 for each correct test case (+24pts total)
    - +2 for correct postfix expression (i.e. correctly implemented method)
        - Flat -4 for returning postfix expression direct from input
    - +1 for correct prefix expression
    - +1 for correct infix expression (Not correct output if it doesn't have parenthesis)
    - +2 for correctly evaluated expression
    - Example of bad parentheses: (((4)+)5) (excessive parentheses are fine if they evaluate, but to my knowledge you cannot evaluate (4 +)
    - Universal -2 for output with no spacing

Prefix	Infix	Postfix	Eval
4	4 or (4)	4	4
+ 4 10	(4+10)	4 10 +	14
* + 2 3 / 8 4	((2+3)*(8/4))	2 3 + 8 4 / *	10
* + 2 - 3 1 / 8 4	((2+(3-1))*(8/4))	2 3 1 - + 8 4 / *	8

Problem 2 (30)

- +1 class correctly named Problem2
- +2 filename taken as command line argument
- +2 File is parsed correctly in Problem2.java
- +4 Does not store a word more than once
- +1 Has UnderFlowException.java
- -10 turns all numbers to empty string
- -1 if numbers not toLowerCase correctly (two and twO different groups)
- Rounding up if a fractional amount of points
- -5 if line number setup incorrectly (can be fixed quickly)
- OK if start line numbering at 0 or 1
- -0.5 for each casse with duplicate numbers listed in linked list
- +0.5 for each correct result
    - tester.txt
    - No points awarded if AVL trees are not used
- -6 PrintIndex not used in Problem2.java

Tester.txt (what we're testing with)

1 hamilton
2 one
3 two
4 three three
5
6
7 One one
8 Two two
9 Three three
10 One twO thrEe
11
12
13 This. children learn a controlled, fun
14 algorithms like GCD, or Factorials. more a

15 ,,creativesa
16 controlled. a
17 applicationss, a
18 applicationsss, a
19 like a gcd
20 purple

Punctuation handling requirements: All lowercase, All punctuation removed and replaced by empty string.
Ex: "..Contr.olled" will be "controlled"

NOTE: no points will be lost if student doesn't take out punctuation for the public List getLinesForWord(String word)

Expected Output (from our solution):
1 [1]
10 [10]
11 [11]
12 [12]
13 [13]
14 [14]
15 [15]
16 [16]
17 [17]
18 [18]
19 [19]
2 [2]
20 [20]
3 [3]
4 [4]
5 [5]
6 [6]
7 [7]
8 [8]
9 [9]
a [13, 14, 16, 17, 18, 19]
algorithms [14]
applicationss [17]
applicationsss [18]
children [13]
controlled [13, 16]