

Assignment 4

Due: 11:59pm April 10th (Tue.)

This assignment is done individually or by a group of 2. Each group submits only 1 copy of the assignment.

README – 2 points

Input/output format (including the function name and the number of arguments) – 3 points

1. **[15 points]** Write a Haskell function **union lt1 lt2** that returns the union of two lists lt1 and lt2. **The resulting list does not contain duplicate elements.**

E.g. `union [1,2,3] [1,2,5] = [3,1,2,5]` (or `[1,2,3,5]` or `[1,2,5,3]`,) // order of elements does not matter

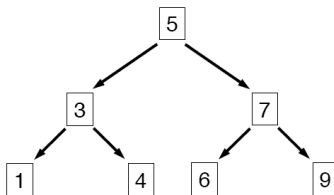
2. **[20 points]** Write a Haskell function **delete k lt** that removes **every kth** element of a list lt.

E.g. `delete 2 [3,4,5,6,7,8,9] = [3,5,7,9]` //remove the 2th, the 4th, and the 6th element

3 **[20 points]** Consider the binary tree defined below:

data Tree = Leaf Int | Node Tree Int Tree

E.g., `Node (Node (Leaf 1) 3 (Leaf 4)) 5 (Node (Leaf 6) 7 (Leaf 9))` specifies the tree:



Define a function **occurs x bt** that decides if an integer x occurs in a binary tree bt.

E.g. `> occurs 9 (Node (Node (Leaf 1) 3 (Leaf 4)) 5 (Node (Leaf 6) 7 (Leaf 9)))`

True

4. **[20 points]** Write a Haskell function **delete_last x lt** to delete the **last** occurrence of x in list lt. If lt does not contain x, then return lt.

E.g. `delete_last 2 [1,2,3,2,4,2,5] = [1,2,3,2,4,5]` // remove the last occurrence of 2

5. **[20 points]** Write a Haskell function **isort lt** that sorts an integer list lt into ascending order using the insertion sort.

E.g. `<isort [7,3,9,2]`
`[2,3,7,9]`

Instruction of Submission:

- Write a program **assignment4.hs** which contains the implementation of all functions.
- Write a **README** file (text file, do not submit a .doc file) which contains
 - Your name and email address.
 - Whether your code was tested on bingsuns.
 - (Optional) Briefly describe anything special about your submission that the TA should take note of.
- Place assignment4.hs and README under one directory with a unique name (such as [userid]_4 for assignment 1, e.g. pyang_4).

- Tar the contents of this directory using the following command.
tar -cvf [directory_name].tar [directory_name]
E.g. tar -cvf pyang_4.tar pyang_4/
- Upload the tared file you created above on the blackboard.

[Academic Honesty:](#)

All students should follow [Student Academic Honesty Code](http://watson.binghamton.edu/acadhonorcode.html) (<http://watson.binghamton.edu/acadhonorcode.html>). All forms of cheating will be treated with utmost seriousness. You may discuss the problems with other students, however, you must write your OWN codes and solutions. Discussing solutions to the problem is NOT acceptable. Copying an assignment from another student or allowing another student to copy your work may lead to an automatic **F** for this course. If you borrow small parts of code/text from Internet, you must acknowledge this in your submission. Also, you must clearly understand and be able to explain the material. Copying entire material or large parts of such material from the Internet will be considered academic dishonesty. Moss will be used to detect plagiarism in programming assignments. You need ensure that your code and documentation are protected and not accessible to other students. Use **chmod 700** command to change the permissions of your working directories before you start working on the assignments. If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.