

Homework 2 (20 pts)

General Instructions

- Due on 12/11 (Sun), 11:59PM
 - Late submission will get no points
- If you cannot finish all, submit solutions for some problems to get partial credits.
- Only submit source files. Do NOT include any executables. All files should be packed into a single zip file and be submitted via blackboard.
- Ensure your code can be compiled and executed in command line (not a java IDE). Otherwise, you will NOT get any credits.
- In the zip file, include a text file: README.txt. Write down which problems have you finished. Also write down on which platform (mac, linux, window) is the code compiled and executed.
- Do not change the main file: HW2MainClass.java.
- Submissions without a proper README or with compiling issues will lose a point.
- This is NOT something you can finish in three days. To understand the problem itself takes quite some time. You have to start as early as possible.
- See the TestData folder for example input and output files.

Problem 1: Generate all possible numbers (6 pts)

- Given an array of positive integers, a , generate all possible arrays with the same length so that at each index, i , the generated number is non-negative and is smaller than the given number, $a[i]$.
- For example, if the input is $[2, 2, 2]$, the output is all possible length-three binary numbers. In general there are $\prod a[i]$ many output. If $a[i] = 0$, the corresponding digit could only be 0.
- Fill in the function Problem1 in Prob1Prob2Class.java, so that your code can be run by calling
 - Java HW2MainClass TEST_P1 input-file

Problem 2: Generating all maximal increasing subsequence (7 pts)

- Given an input array, a maximal increasing subsequence is a subsequence whose values are increasing, and adding any other element in the array will not be an increasing subsequence.
- We assume all numbers in the array are distinct.
- For example, if the input is $[0, -10, 10, 5, 20]$, the output will be $[-10,5,20]$, $[-10,10,20]$, $[0,5,20]$ and $[0,10,20]$
- You cannot generate all possible sequences and then screen them (you will lose points).
- You have to make the smart decision in choosing the good first/second/third ... last entries. Think about what numbers are eligible once previous decisions have been made.
- Fill in function Problem2 in Prob1Prob2Class, so that your code can be run by calling
 - Java HW2MainClass TEST_P2 input-file

Problem 3: Generating Balanced Trees (7 pts)

- For a given array, and a tolerance, generate all possible binary trees with the same inorder traversal, so that at any node, its left and right subtree heights differ by no more than the tolerance. Output all the preorder traversals.
- We assume all numbers in the input array are distinct.
- For example, if the input is [1, 2, 3, 4] and the tolerance is 1, the output will be [2,1,3,4], [2,1,4,3], [3,1,2,4], [3,2,1,4].
- You cannot generate all possible trees and screen them (you will lose points).
- Use recursion. At each node, first generate all possible left subtrees, then generate all possible right subtrees so that its height is within +/- tolerance of the left subtree height.
- Finish the class BinaryTreeClass, so that your code can be run by calling
 - Java HW2MainClass TEST_P3 input-file

Additional tools: generating your own scripts

- Generate your own test case using MyRandGeneratorClass.
- To see how, try
 - java HW2MainClass INPUT_P1 input-file-prefix
 - java HW2MainClass INPUT_P2 input-file-prefix
 - java HW2MainClass INPUT_P3 input-file-prefix
- Each command will generate 10 test files using the given prefix.
- TestAll.py – a python script to run through all test files in TestData folder. Note that it will overwrite the existing output files.