Assignment 3

General Instructions

You have to do this assignment as a **team of two students** if you are an undergraduate student, or **individually** if you are a postgraduate student. Individual work of an undergraduate student is also allowed and will be treated same as team work. Team members should be from the same tutorial group. All implementations have to be done in JAVA.

Submissions have to include coversheet including names, student ids, and your tutorial group such that submissions can get marked.

Submit your solutions (including printout of the source code) for Exercises 1, 2, and 3 as well as the results of the execution of your programs to the box ”ADSA” on level 4, Innova21 (close to reception) **by the deadline**. **No late submissions will be accepted.**

In addition, submit your source code for Exercises 1 and 3 using the websubmission system.

Exercise 1 *Counting binary search trees (2 + 3 + 2 points)*

We are given a set of N keys { 1, 2, ..., N}.

1. How many unique binary search trees (BSTs) can be created from the N keys?
2. Design and implement an algorithm that counts the number of BSTs for input N. Test your algorithms for $N = 2, 4, 8, 16, 32, 64, 128, 256$.
3. Show your algorithm runs in the worst-case time complexity of $O(N^2)$.

Exercise 2 *Building heaps (CLRS Problem 6.2) (3+ 3 points)*

We can build a heap by repeated calling MAX-HEAP-INSEART to insert the element into the heap. Consider the folloing variation of BUILD-MAX-HEAP procedure:

```
BUILD-MAX-HEAP'($A$)
$A.heap\text{-}size = 1$
for $i = 2$ to $A.length$
    MAX-HEAP-INSEART($A, A[i]$)
```

1. Do the procedure BUILD-MAX-HEAP and BUILD-MAX-HEAP' always create the same heap when run on the same input array? Prove they do, or provide a counter-example.
2. Show that in the worst case, BUILD-MAX-HEAP' requires $\Theta(n \log n)$ time to build an n-element heap.

Exercise 3 *Searching in sorted matrix (CLRS Problem 6-3) (5+2 points)*

You are given an $m \times n$ matrix of numbers A such that the entries of each row are in sorted order from left to right and the entries of each cloumn are in sorted order from top to bottom.

1. Design and implement an algorithm that finds the location ((i, j) such that $A[i, j] = x$) of an arbitrary value x in the matrix or report that the item is not present. Your algorithm should run in $O(m + n)$ time. Test your program for finding 1, 3, 16, 18, 25, 28, 31, 32, 40 in the following matrices:

$$(a) \begin{pmatrix} 2 & 5 & 7 & 9 & 11 \\ 4 & 12 & 13 & 22 & 23 \\ 6 & 14 & 15 & 24 & 27 \\ 8 & 16 & 19 & 26 & 29 \\ 19 & 18 & 21 & 28 & 31 \end{pmatrix} \quad (b) \begin{pmatrix} 2 & 5 & 7 & 9 & 11 & 12 & 14 & 15 & 19 & 21 \\ 4 & 6 & 8 & 10 & 12 & 13 & 15 & 17 & 20 & 22 \\ 6 & 7 & 9 & 11 & 13 & 14 & 17 & 19 & 21 & 24 \\ 7 & 8 & 10 & 12 & 14 & 15 & 18 & 20 & 22 & 26 \\ 8 & 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 & 27 \\ 9 & 10 & 12 & 13 & 16 & 19 & 20 & 22 & 24 & 28 \\ 10 & 11 & 13 & 14 & 17 & 20 & 21 & 23 & 26 & 29 \\ 21 & 22 & 24 & 26 & 27 & 29 & 30 & 33 & 34 & 36 \\ 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 \end{pmatrix}$$

2. Analyse the time complexity of your algorithm.

Submission instructions for programming code

First, type the following command, all on one line (replacing `aXXXXXXX` with your username):

```
svn mkdir --parents -m "ADSA"
https://version-control.adelaide.edu.au/svn/aXXXXXXX/2014/s2/adsa/assignment3
```

Then, check out this directory and add your files:

```
svn co https://version-control.adelaide.edu.au/svn/aXXXXXXX/2014/s2/adsa/assignment3
cd assignment3
svn add File1.java
svn add File2.java
...
svn commit -m "assignment3 solution"
```

Next, go to the web submission system at:

https://cs.adelaide.edu.au/services/websubmission/

Navigate to *2014*, *Semester 2, Adelaide*, *Algorithm and Data Structure Analysis*, then *Assignment 3*. Click *Make A New Submission For This Assignment* and indicate that you agree to the declaration. The automark script will then check whether your code compiles. You can make as many resubmissions as you like. If your final solution does not compile you won't get any marks for this solution.

The websubmission system for this assignment will open till 10am Monday September 1st, 2014.

End of Questions