CSC 340 HOMEWORK 1 /QUIZ 1 STUDY GUIDE

PROFESSOR GODFREY MUGANDA

1. Algorithm Design

These two programs are due Friday night of week 2 at midnight.

1. Create a single Netbeans project whose name is Homework1 prefixed by your last name, as in MugandaHomework1. As you are creating the project, uncheck the option to create a main class.

Next, under source packages, right click on the default package and select

`New --- Java Class.`

Name the class `Problem1`.

Add a main method to the `Problem1` class, and write a program which when run, asks the user to enter two positive integers n, m where $n \leq m$. The program then creates a list or array of m integers all of which are less than 100. The program prints these numbers out, followed by a list of the n smallest of the m numbers on the next line. Here is a sample run:

```
Enter two positive inters n m where n <= m:
5 10
Here are 10 random integers:
[5, 60, 99, 14, 70, 3, 27, 62, 1, 79]
Here are the 5 smallest integers:
[1, 3, 5, 14, 27]
```

You can run the program by right-clicking on the file name `Problem1.java` node under default package and selecting `Run File`.

2. To the same Netbeans project, right-click the default package and and add a new Java class called `Problem2`.

Write a program that asks the user to enter a positive integer $n \geq 3$. The program then prints all subsets of size 3 of the set $\{1, 2, \ldots, n\}$. The element each subset of size 3 should be printed on its own line. You may use any format to print: separated by commas, or spaces, enclosed in brackets, whatever.

Here is a couple of sample runs. (You can run the file by right-clicking on the file name and selecting `Run File`.)

```
Enter a positive integer greater than 2:
5
Subsets of size 3 of 1.. 5:
1, 2, 3
1, 2, 4
1, 2, 5
1, 3, 4
```

```
1, 3, 5
1, 4, 5
2, 3, 4
2, 3, 5
2, 4, 5
3, 4, 5

Enter a positive integer greater than 2:
3
Subsets of size 3 of 1.. 3:
1, 2, 3
```

2. Concepts for Quiz 1

You need to understand the following important terms: algorithm, computational problem, efficiency of an algorithm, time and space complexity functions, basic step or basic operation, input size of a problem, worst-case and average-case complexity.

You should be able to solve recurrence relations such as solved in class.

As an exercise (Does not have to be handed in) do problem 14 on page 90 in the class text.