60-100 Individual assignment #10 to hand in on Thursday 27th or Friday 28th November 2014.
PART I

The Miranda programming environment has a "switch" that can be used to count the number of basic operations, and space, used when executing a program. The basic operations are called "reductions" and can be used as an approximate time cost of executing the program. The switch can be activated by typing `/count` at the Miranda prompt.

The time cost of running a program `p` for different inputs, can be estimated as follows:

1) Put the program `p` in a file `p.m` (with other programs if you wish)
2) Start Miranda by typing `mira p.m`
3) Type `/count` at the Miranda prompt.
4) Run the program p with different inputs. For each input record the number of reductions (which will be displayed after the program has finished executing).

Here is an example, using the Fibonacci program `fib` (from the lectures):

```
fib 0 = 1
fib 1 = 1
fib n = fib (n – 1) + fib (n – 2)
```

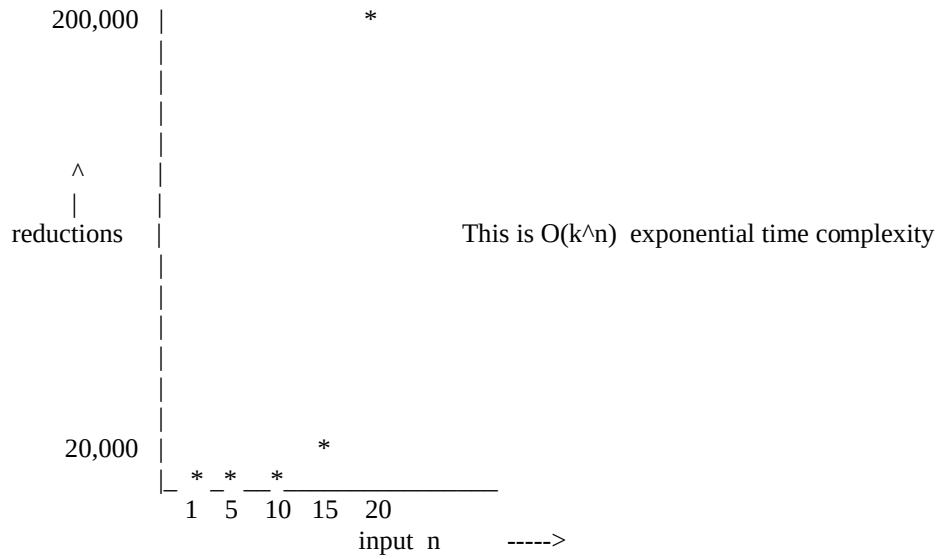
If this program were in a file called `fib.m`, then:

`mira fib.m` - this starts the Miranda environment with file `fib.m`

```
Miranda /count

Miranda fib 1
1
||reductions = 6, cells claimed = 10, no of gc's = 0, cpu = 0.00
Miranda fib 4
5
||reductions = 88, cells claimed = 92, no of gc's = 0, cpu = 0.00
Miranda fib 5
8
||reductions = 153, cells claimed = 155, no of gc's = 0, cpu = 0.00
Miranda fib 10
89
||reductions = 1892, cells claimed = 1845, no of gc's = 0, cpu = 0.00
Miranda fib 15
987
||reductions = 21174, cells claimed = 20573, no of gc's = 0, cpu =
0.00
Miranda fib 20
10946
||reductions = 235015, cells claimed = 228261, no of gc's = 0, cpu = 0.0
```

If we create a graph with vertical axis “reductions” and horizontal axis “input n”, we get the following TIME COMPLEXITY graph (approximately):



PART II

Copy each of the programs given in Question #9 of the Final Exams for 2008, 2009, 2010, 2011, 2012 and Winter 2014 (see class notes) into a file called assign_10.m, all the programs p_2008, p_2009, etc. Now record the number of reductions for 6 inputs (see below) for each program, and plot the number of reductions (vertical axis) against the size of the input (horizontal axis). Try to hand-fit a curve for each graph. Try to guess the TIME COMPLEXITY of each program from the curve. (For example, if you plotted the data above for the Fibonacci program, and fitted a curve, you might guess, correctly, that the time complexity is exponential.) Note that your graphs can be “rough” and do not have to be exact.

Note: You should choose the 6 inputs by finding a large input say “big” (which will differ for each program) which takes a few seconds to return an answer. Then choose 6 values between 1 and “big” as the input values for your graph.