PROBLEM SET 03

YOUR NAME

(1) Let f, g, and h all be bijective functions (so that they have inverses) and they all map real numbers to real numbers.

(a) What is the inverse of $f \circ (g \circ h)$ in terms of f^{-1}, g^{-1}, and h^{-1}?

(b) Prove or disprove that $f + g + h$ is always bijective.

(2) This question deals with the *Mystery* function which takes in an ordered pair of 2 integers greater than 0, does the following algorithm, and returns a single integer i.

$Mystery\,(\langle n, m \rangle)$
$i := 0$
while $n > 0$
    $p := m$
    $n := n - 1$
    while $p > 0$
        $i := i + 1$
        $p := p - 1$
return i

(a) In more common language, what does *Mystery* do? I.e., what is its return value, i, in terms of n and m?

(b) What's the big O runtime complexity of *Mystery* in terms of n and m?

(c) What's the best runtime an algorithm can have to compute the *Mystery* function?

(3) Curious George has decided to answer his problem set 3 by doing the following algorithm:

- George opens ps03.tex in his text editor and randomly presses keys on his laptop for several minutes (some of the keys (like alt) do nothing, and some, like the arrow keys, move the cursor around).

- George then runs LaTeX and compiles his program. If he gets an error, he deletes everything and starts over.
- If his LaTeX run is successful, George prints his "solution set" out and takes it to HouCheng, who grades it for him almost instantly.
- If George's score is less than perfect, George starts over.

(a) What is the *average* big O runtime complexity of George's algorithm in terms of n, the number of characters in The Perfect Solutions Set to Problem Set 3?

(b) What's the *best case* complexity?

(c) What's the *worst case* complexity?

(d) If George types 10 characters a second, about how long would it take him to produce every sequence that's as long as the phrase

`just_over_one_decillion_years`

if his laptop has 27 characters (26 letters plus the space "_")?

Note that among other sequences, George would type:

```
slfjqz_nnper_nnuiv__sfns_sdfl
sdc__qpirun_zocnweiufn_njf___
_sdc_qpirun__zocnweiufn_njf__
_monkey___hate__type_laptop__
all_work_and_no_play_make_geo
rge_dull_monkey_all_work_and_
just_over_one_decillion_years
```

Please also note that I made an error in lecture: the term (in America, not England) for 10^{33} is *decillion*, **not** dectillion.

(4) BONUS: The *Ackermann* function maps $N \times N$ to N, where N is the set of positive integers.

- $A(\langle 0, n\rangle) = n + 1$
- $A(\langle m, 0\rangle) = A(\langle m-1, 1\rangle)$ if $m > 0$

- $A(\langle m, n\rangle) = A(\langle m-1, A(\langle m, n-1\rangle)\rangle)$ if $m > 0$ and $n > 0$

(a) What is $A(\langle 2, 2\rangle)$?

(b) How many digits long is the decimal form of $A(\langle 4, 2\rangle)$? (To get credit, you can either write the answer, or write pseudo-code for an algorithm that computes the Ackermann function. To save trees, please *do not* write out the full decimal form of $A(\langle 4, 2\rangle)$.)

(Also, if you thought Ackermann's function grew quickly, check out

`http://www.scottaaronson.com/writings/bignumbers.html`

My favorite is the Busy Beaver.)