cmput 304 2023 study questions 1

1. Recall `http://webdocs.cs.ualberta.ca/~hayward/272/jem/collatz.html`. Prove or disprove (if you can):

   a) for all positive integers n, the last number printed by `collatz(n)` is 1.

   b) for all n in $\{1,2,\ldots,10\}$, the last number printed by `collatz(n)` is 1.

   c) if the collatz conjecture fails for some integer, and if n_0 is the smallest such integer, then n_0 is odd.

2. Use the old-school square root algorithm from class: by hand, find the largest integer $t \leq \sqrt{762309}$. Show your work. Check your answer:

   `https://github.com/ryanbhayward/algs/blob/master/numeric/old_sqrt.py`

3. a) For an input of size (number of bits) n, an algorithm has runtime $\Theta(n^{2.5})$. For $n = 3.7e6$ (scientific notation: 3.7×10^6), the runtime is 11s. What is your best guess for the runtime when $n = 7.4e6$? Explain briefly. Show your work.

   b) Is it possible that your guess in a) will be 10 times too small? Explain briefly.

4. Let $f(n) = 2^n$ and $g(n) = 1.7^n$. Prove or disprove: $f(n) \in O(g(n))$.

5. Recall iterative Fibonacci:

   `https://webdocs.cs.ualberta.ca/~hayward/304/jem/warmup.html#ifib`

   a) Explain why the runtime of `ifib(n)` is proportional to $\sum_{j=1}^{n} \lg(\text{fib}(j))$.

   b) Explain why the above sum is in $O(n^2)$.

   c) Let $r(n)$ be the runtime for `ifib(n)`. For a positive integer t, explain why we expect that $r(2^{t+1})/r(2^t)$ will be around 4.

6. Recall recursive `fib(n)`:

```
def fib(n):
  if (n<=1): return n
  return fib(n-1) + fib(n-2)
```

   Complete the proof of the following claim.

   Claim. For all integers $n \geq 0$, `fib(n)` returns $f(n)$, where $f(n)$ is defined as 0 if n is 0, 1 if n is 1, and the sum of $f(n-1)$ and $f(n-2)$ when n is at least 2.

   Proof. Argue by induction on n. The claim holds when n is 0 or 1 (why?).

   Let x be an integer greater than or equal to 2. Assume that the claim holds for all values of n in the set $\{0, 1, \ldots, x-1\}$. In order to complete the proof, we now want to show that the claim holds when n is x, i.e. that `fib(x)` returns $f(x)$.

   So what happens when `fib(x)` executes? Well, $x \geq 2$, so the `if` test evaluates to false, so the program returns `fib(x-1)+fib(x-2)`.

   (now finish the proof ...)

7. For non-negative integers x, y with $x < y$, prove that the runtime of the usual addition algorithm takes time $\Theta(\lg y)$.

8. For non-negative integers x, y with $x < y$, prove that the runtime of any addition algorithm takes time $\Omega(\lg y)$.

9. Repeat question 3 for runtime $\Theta(n^2)$.

10. Repeat question 3 for runtime $\Theta(n \lg n)$.