CS 131 – Fall 2019, Assignment 7

Problems must be submitted by Friday November 1, 2019 5:00pm, on Gradescope.

Problem 1. Use the Master theorem to give upper bounds for the following functions.

a) $f(n) = f(n/4) + 2n^{1/2}$

b) $f(n) = 3f(n/4) + 2n^{1/2}$

Problem 2. Consider the algorithm below which computes $\lfloor a^{1/b} \rfloor$ where a and b are positive integers.

function SEARCH(a, b, l, r)
    if $l = r$ **then**
        return l
    end if
    $m \leftarrow \lceil \frac{l+r}{2} \rceil$
    if $m^b \leq a$ **then**
        return SEARCH(a, b, m, r)
    else
        return SEARCH$(a, b, l, m-1)$
    end if
end function
function ROOT(a, b)
    return SEARCH$(a, b, 1, a)$
end function

a) List all calls to SEARCH (direct or indirect) that ROOT$(150, 3)$ makes. How many are there?

b) Use the Master theorem to upper bound the running time of ROOT(a, b). Assume a is a power of 2.

Problem 3.

a) Convert 303 to binary.

b) Convert 303 to base 3.

c) Convert 1001011_2 to decimal.

d) Convert 12012_3 to decimal.

e) What is the last digit of $123456789012345678 \times 609$ in base 6. Don't try to compute the product.

Problem 4. Let f_n be Fibonacci numbers with $f_0 = 0$, $f_1 = 1$. This problem asks you to prove that any positive integer can be uniquely represented as a sum of Fibonacci numbers $f_2, f_3, f_4, \ldots$ with no repetitions and no two consecutive Fibonacci numbers.

For instance, $20 = f_7 + f_5 + f_3$.

Thus, Fibonacci numbers form a peculiar number system.

a) Represent the following numbers as explained above: $1, 7, 14, 88, 199$.

b) Prove using induction that any positive integer can be represented as a sum of Fibonacci numbers $f_2, f_3, f_4, \ldots$, possibly with repetitions or consecutive Fibonacci numbers.

c) Now prove using strong induction that any positive integer can be represented as a sum of Fibonacci numbers $f_2, f_3, f_4, \ldots$, with no consecutive Fibonacci numbers and no repetitions.

d) (Extra credit for 15%) Prove that for any positive integer its representation as a sum of Fibonacci numbers $f_2, f_3, f_4, \ldots$ with no consecutive Fibonacci numbers and no repetitions is unique.

Problem 5. Recall that for non-negative integers a and b, $\gcd(a, b)$ is the largest integer that divides both a and b. For instance, $\gcd(108, 60) = 12$ since $12|108$ and $12|60$ and there is no larger integer that divides both. To see this, we can perform the algorithm that humans most like: decompose a and b into products of primes and pick as many primes from both as possible. For instance,

$$108 = 2 \cdot 2 \cdot 3 \cdot 3 \cdot 3$$
$$60 = 2 \cdot 2 \cdot 3 \cdot 5$$

We can pick at most two 2's and one 3 from both. $\gcd(108, 60) = 2 \cdot 2 \cdot 3 = 12$.

This algorithm is convenient and works well for small numbers, but factoring is very inefficient when large numbers are involved. Below is an implementation of Euclid's algorithm for gcd which needs fewer steps on large inputs.

```
function EUCLID(a, b)
    if a = 0 then
        return b
    end if
    if b = 0 then
        return a
    end if
    if a ≤ b then
        return EUCLID(a, b mod a)
    else
        return EUCLID(a mod b, b)
    end if
end function
```

a) Run $\text{EUCLID}(108, 60)$ and list all recursive calls.

b) Prove that if a and b are non-negative integers, $\gcd(a, b) = \gcd(a, b - a)$.

As a consequence, $\gcd(a, b) = \gcd(a, b \mod a)$. To see this one can use the fact that $\gcd(a, b) = \gcd(a, b - ak)$ for any integer k which can be proven the same way the claim in the previous part is proven. Now, $b \mod a$ is simply $b - ak$ for some integer k. So, $\gcd(a, b \mod a) = \gcd(a, b - ak) = \gcd(a, b)$.

c) Prove correctness of EUCLID. That is, show that

$$\forall a, b(\neg(a = 0 \wedge b = 0) \rightarrow (\text{EUCLID}(a, b) = \gcd(a, b))).$$

To do that, for all $n \geq 0$ define predicate $P(n)$ to be true if and only if

$$\forall a, b(\neg(a = 0 \wedge b = 0) \wedge a \leq n \wedge b \leq n \rightarrow (\text{EUCLID}(a, b) = \gcd(a, b))).$$

Prove by induction that $P(n)$ is true for all $n \geq 0$ and conclude the proof.