CSCI 2300: Introduction to Algorithms
Homework 2

Lucien Brule
Prof. Bulent Yener
April 27, 2023

1 Problem 1

Problem: Give an algorithm (pseudo code, with explanation) to compute 2^{2^n} in linear time, assuming multiplication of arbitrary size integers takes unit time. What is the bit-complexity if multiplications do not take unit time, but are a function of the bit-length?

Part A: Algorithm

```
function power_of_two(n):
    result = 1
    for i in range(0, 2^n):
        result *= 2
    return result

function compute_2_2_n(n):
    return power_of_two(power_of_two(n))
```

Explanation: The function *power_of_two(n)* computes 2^n by initializing the result to 1 and then multiplying it by 2 for n times. In the *compute_2_2_n(n)* function, we first compute 2^n using the *power_of_two(n)* function and then pass the result back to the same function to compute 2^{2^n}.

Part B: Bit-Complexity

When multiplication of arbitrary size integers does not take unit time, the complexity of multiplication becomes a function of the bit-length. Let's assume the bit-complexity of multiplying two n-bit integers is $O(M(n))$.

In the *power_of_two(n)* function, we perform $2^n - 1$ multiplications, and the size of the integer doubles with each multiplication. Therefore, the bit-complexity of this function is:

$$O\left(\sum_{i=1}^{2^n} M\left(2^{i-1}\right)\right)$$

Since we call the *power_of_two()* function twice in the *compute_2_2_n()* function, the total bit-complexity for the entire algorithm is:

$$O\left(\sum_{i=1}^{2^n} M\left(2^{i-1}\right) + \sum_{i=1}^{2^{2^n}} M\left(2^{i-1}\right)\right)$$

This is the bit-complexity of the algorithm for computing 2^{2^n} when the multiplication cost depends on the bit-length.

2 Problem 2

Problem: Consider the problem of computing $N! = 1 \cdot 2 \cdot 3 \cdots N$.
(a) If N is an n-bit number, how many bits long is $N!$ in $O()$ notation (give the tightest bound)?
(b) Give an algorithm to compute $N!$ and analyze its running time.

Part A:

To find the number of bits in $N!$, we can use Stirling's approximation, which states:

$$N! \approx \sqrt{2\pi N}\left(\frac{N}{e}\right)^N$$

Taking the logarithm base 2 of both sides, we get:

$$\log_2(N!) \approx \log_2(\sqrt{2\pi N}) + N\log_2\left(\frac{N}{e}\right)$$

Since $\log_2(\sqrt{2\pi N})$ is $O(\log N)$ and $N\log_2\left(\frac{N}{e}\right)$ is $O(N\log N)$, the tightest bound for the number of bits in $N!$ is:

$$O(N\log N)$$

Part B: Algorithm and Running Time

```
function factorial(N):
    if N == 0 or N == 1:
        return 1
    else:
        return N * factorial(N - 1)
```

Explanation: The function *factorial(N)* computes the factorial of N using a recursive approach. If N is 0 or 1, the function returns 1, as $0! = 1! = 1$. Otherwise, it computes the factorial by multiplying N with the factorial of $(N-1)$, obtained by calling the function recursively.

Running Time Analysis: The function *factorial(N)* is called recursively N times, and each function call involves one multiplication operation. Assuming multiplication of arbitrary size integers takes unit time, the running time of this algorithm is $O(N)$.

However, if we consider the bit complexity of multiplication, the running time will be different. If we assume the multiplication of two n-bit integers takes $O(M(n))$ time, then the running time of the algorithm will be:

$$O\left(\sum_{i=1}^{N} M(i)\right)$$

3 Problem 3

Problem: Find the GCD of 1492 and 1776, using
(a) the prime factorization method and using Euclid's method, and
(b) express the GCD as an integer linear combination of the two inputs.

Solution:

Part A:

Prime factorization method:

$1492 = 2 \times 2 \times 373 = 2^2 373$
$1776 = 222337 = 2^3 337$

The GCD is the product of the common prime factors with the lowest exponent:

GCD(1492, 1776) $= 2 \times 2 = 4$

Euclid's method:

gcd(1492, 1776): $1776 = 1492 \times 1 + 284$ $1492 = 284 \times 5 + 40$ $284 = 40 \times 7 + 4$ $40 = 4 \times 10$

The last non-zero remainder is 4. Therefore:

GCD(1492, 1776) = 4

Part B:

Using the extended Euclidean algorithm:

gcd(1492, 1776): 4 = 284 - 40 $\times$ 7 = 284 - (1492 - 284 $\times$ 5) $\times$ 7 = 284 $\times$ 36 - 1492 $\times$ 7 = (1776 - 1492 $\times$ 1) $\times$ 36 - 1492 $\times$ 7 = 1776 $\times$ 36 - 1492 $\times$ 37

The GCD as an integer linear combination of 1492 and 1776:

GCD(1492, 1776) = 1776 $\times$ 36 - 1492 $\times$ 37 = 4