Week 1 Studio Sheet

(To complete at home during week 1)

Objectives: These problems aim to assess your mathematical background and preparedness for FIT2004. If the techniques required to solve these problems are foreign to you, please attend a consultation and plan to catch up on whatever you are missing. The tools and techniques demonstrated by these problems will be necessary to solve problems for the remainder of the unit, so do not fall behind.

Problem 1. Show, using elementary properties of the logarithm function, that the following identities are true

(a) $\log_2\left(\frac{k+1}{2}\right)+1=\log_2(k+1)$

(b) $a^{\log_b(n)}=n^{\log_b(a)}$ for any base $b>1$

Problem 2. Using mathematical induction, prove the following identity:

$$\sum_{i=1}^{n} i \coloneqq 1+2+\ldots+n=\frac{n(n+1)}{2}, \qquad \text{for } n\geq 1.$$

Problem 3. Using mathematical induction, prove the following identity:

$$\sum_{i=0}^{n} r^i \coloneqq 1+r+r^2+r^3+\ldots+r^n=\frac{r^{n+1}-1}{r-1}, \qquad \text{for all } n\geq 0, r\neq 1.$$

Problem 4. Using Problem 3, show that the following inequality is true for all integers $n\geq 1$

$$\sum_{i=0}^{n} \frac{1}{2^i} \coloneqq 1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots+\frac{1}{2^n}<2$$

Problem 5. The harmonic numbers are defined as the partial sums of the harmonic sequence:

$$H(n) \coloneqq \sum_{i=1}^{n} \frac{1}{i}=1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}$$

We want to determine how this quantity behaves asymptotically, as it frequently arises in algorithm analysis.

(a) Prove that $H(n)>\log_e(n)$

(b) Prove that $H(n)\leq \log_e(n)+1$

(c) Hence deduce that $H(n)=\Theta(\log(n))$ [1]

[Hint: Consider the area under the function $f(x)=\frac{1}{x}$ using calculus and compare this with $H(n)$ geometrically.]

Problem 6. Prove that for all $N\geq 0$,

$$\sum_{i=0}^{N} 2^i = 2^{N+1}-1.$$

Problem 7. For each of the following expressions, write a tight upper bound as $n\to\infty$ in big-O notation:

[1] Remember that the notation $\Theta(f(n))$ means that $H(n)$ is both upper bounded by $c_1 f(n)$ for some c_1 (it is $O(f(n))$ and lower bounded by $c_2 f(n)$ for some c_2 (it is $\Omega(f(n))$).

(a) $3n^3 + 100n^2 + n$

(b) $n^3 \log(n) + 0.5n^4 + 100n^2$

(c) $5\sqrt{n} + 10\log(n)$

(d) $\log(n) + \log(\log(n)) + \log(\log(\log(n)))$

(e) $2n\log(n) + 8n\log(n^2)$

(f) $3n\log(n) + 5n(\log(n))^2$

(g) $\log(n) + 2\log^2(n)$

(h) $3\log(n) + (\log(\log(n))^2$

Problem 8. The Fibonacci sequence is defined by the recurrence relation $F(n) = F(n-1) + F(n-2)$, with $F(1) = F(2) = 1$.

(a) Write a Python function that, given a non negative integer n, computes the n^{th} Fibonacci number iteratively (using a loop). Analyse the time and space complexity of your implementation.

(b) Write a Python function that, given a non negative integer n, computes the n^{th} Fibonacci number recursively. Analyse the time and space complexity of your implementation.

Think carefully about the assumptions that you make when analysing the time and space complexity of your algorithms.