Algorithms

December 18, 2022

1 Prologue

(0.1) In each of the following situations, indicate whether $f = \mathcal{O}(g)$, if $f = \Omega(g)$, or if $f = \Theta(g)$.

	$f(n)$	$g(n)$
(a)	$n - 100$	$n - 200$
(b)	$n^{1/2}$	$n^{2/3}$
(c)	$100n + log(n)$	$n + (log(n))^2$
(d)	$n * log(n)$	$10n * log(10n)$
(e)	$log(2n)$	$log(3n)$
(f)	$10log(n)$	$log(n^2)$
(g)	$n^{1.01}$	$n * log^2(n)$
(h)	$n^2/log(n)$	$n * log^2(n)$
(i)	$n^{0.1}$	$log^{10}(n)$
(j)	$(log(n))^{log(n)}$	$n/log(n)$
(k)	$\sqrt{n}$	$log^3(n)$
(l)	$\sqrt{n}$	$5^{log_2(n)}$
(m)	$n * 2^n$	3^n
(n)	2^n	2^{n+1}
(o)	$n!$	2^n
(p)	$log(n)^{log(n)}$	$2^{(log_2(n)^2}$
(1)	$\sum_{i=1}^{n} i^k$	n^{k+1}

Solution: add solution here

(0.2) Show that, if c is a positive real number, then $g(n) = 1 + c + c^2 + \cdots + c^n$ is

(a) $\Theta(1)$ if $c < 1$
(b) $\Theta(n)$ if $c = 1$
(c) $\Theta(c^n)$ if $c > 1$

Solution: add solution here

(0.3) In this problem we confirm the Fibonacci numbers grow exponentially fast, and obtain bounds on their growth.

(a) Use induction to prove $F_n \geq 2^{0.5n}$ for $n \geq 6$

(b) Find a constant c, such that $F_n \geq 2^{0.5n}$ for $n \geq 6$

(c) What is the largest c you can find, such that $F_n = \Omega(2^{cn})$

Solution:

(0.4) This problem describes a method `fib3` for computing the *nth* fibanacci number using 2×2 matricies. See book for description of the algorithm. We analyze its performance. In the following let

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

(a) Show two 2×2 matricies cam be multiplied using 4 additions and 8 multiplications.

(b) Show $\mathcal{O}(log(n)$ matrix multiplications suffice for computing X^n

(c) Show that all intermediate results of `fib3` are $\mathcal{O}(n)$ bits long

(d) Let $M(n)$ be the running time of an algorithm for multiplying n-bit numbers, and assume the $M(n) = \mathcal{O}(n^2)$. Prove the running time of `fib3` is $\mathcal{O}(M(n)log(n))$

(e) Can you prove the running time of `fib3` is $\mathcal{O}(M(n))$? Assume $M(n) = \Theta(n^a)$ for some $1 \leq a \leq 2$.

Solution: