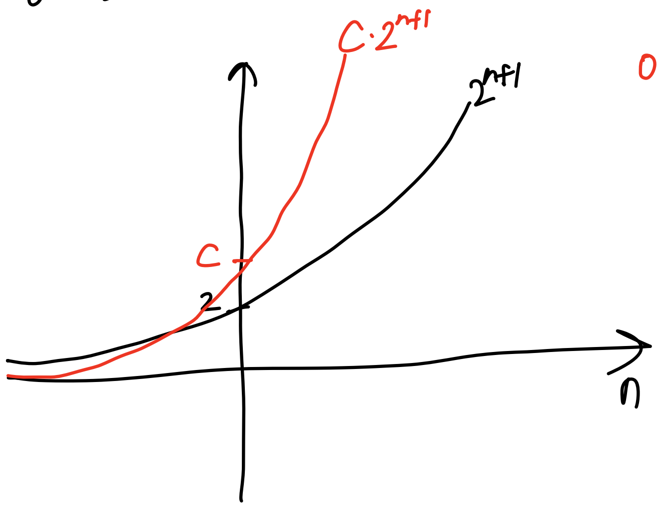
Algorithms analysis	Section	01
	Student number	22000080
Homework 1	Name	Kim, Min Chae

// Print this document and write the solution to each problem below.
// then scan your answer sheet and submit through HisNet assignment board.
// Make sure that Copied answer will not be accepted.

1. $2^{n+1} = O(2^n)$?

if true, there exist positive constants C and n_0 such that

$0 \leq 2^{n+1} \leq C \cdot 2^n$



$0 \leq 2^{n+1} \leq C \cdot 2^n$

with $c=2$ and $n_0 = 0$

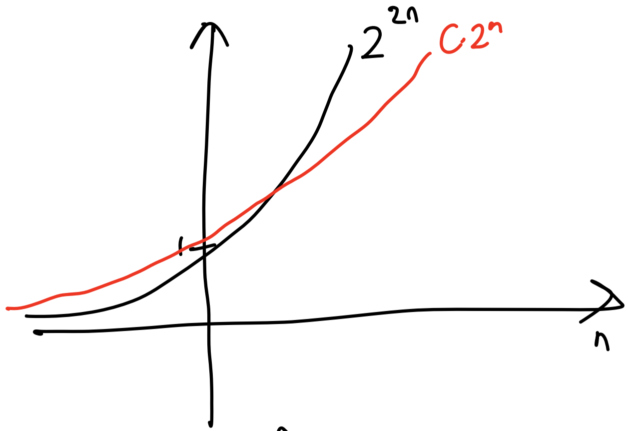
So, it is true

2. $2^{2n} = O(2^n)$?

if true, there exist positive constants C and n_0 such that

$0 \leq 2^{2n} \leq C \cdot 2^n$

not hold true for any c



So, it is false

3. For two functions $f(n)$ and $g(n)$,

we have $f(n) = \Theta(g(n))$ if and only if

$f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

$\Theta(g(n)) = \{ f(n)$: there exsist positive constants

c_1, c_2 and n_0 such that

$\boxed{\boxed{0 \le c_1 g(n) \le f(n)} \le c_2 g(n)}$ for all $n \ge n_0$.

① in this, $0 \le c_1 g(n) \le f(n)$ for all $n \ge n_0$

is definition of Ω-notation.

② in this, $0 \le f(n) \le c_2 g(n)$ for all $n \ge n_0$

is definition of O-notation.

$f(n) = \Theta(g(n))$ should satisfy both ① Ω-notation

and ② O-notation,

so $f(n) = \Theta(g(n))$ if and only if

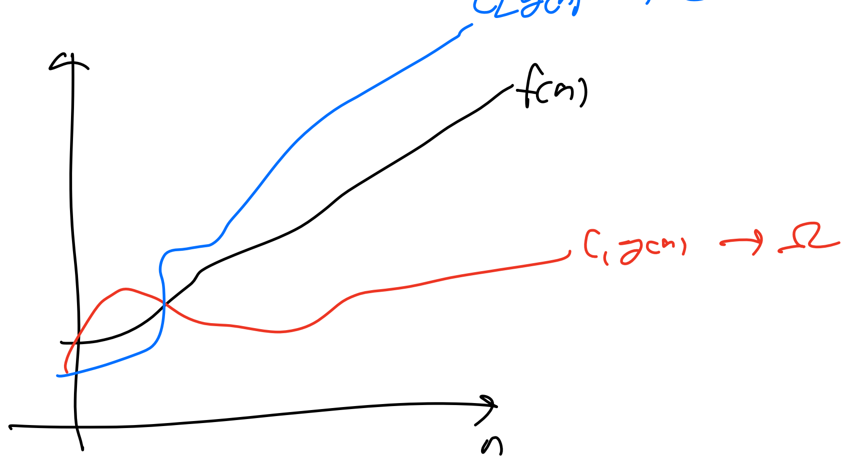
$f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ is true.

4. the running time of an algorithm is $\Theta(g(n))$

if and only if its worst-case running time is $O(g(n))$

and its best-case running time is $\Omega(g(n))$

in Θ-notation,



O-notation gives an upper bound on a function, to within a constant factor.

In other words, it is the worst-case limit that consumes the most time.

so its worst-case running time is $O(g(n))$

Ω-notation gives an lower bound on a function, to within a constant factor.

In other words, it is the best-case limit that consumes the least time.

so its best-case running time is $\Omega(g(n))$

5. $o(g(n)) \cap \omega(g(n))$ is the empty set.

$o(g(n)) = \{ f(n) :$ for any positive constant $c > 0$, there exist a constant $n_0 > 0$ such that $0 \le f(n) < c \cdot g(n)$ for all $n \ge n_0$.

and $\omega(g(n)) = \{ f(n) :$ for any positive constant $c>0$, there exist a constant $n_0 > 0$ such that

$0 \le c \cdot g(n) < f(n)$ for all $n \ge n_0 \}$.

so $O(g(n)) \cap \omega(g(n))$ is should satisfy

$0 \le c \cdot g(n) < f(n) < c \cdot g(n))$ for all $n \ge n_0$, but

It's wrong statement.

So $O(g(n)) \cap \omega(g(n))$ is the empty set.

6. For a given function $g(n,m)$, we denote by $O(g(n,m))$ the set of functions

$O(g(n,m)) = \{ f(n,m) :$ there exist positive constants c, n_0, and m_0 such that

$0 \le f(n,m) \le c \cdot g(n,m)$ for all $n \ge n_0$ and $m \ge m_0 \}$

→ give corresponding definitions for $\Omega(g(n,m))$ and $\Theta(g(n,m))$

$\Omega(g(n,m)) = \{ f(n,m) :$ there exist positive constraints c, n_0, and m_0 such that

$0 \le c \cdot g(n,m) \le f(n,m)$ for all $n \ge n_0$ and $m \ge m_0 \}$

$\Theta(g(n,m)) = \{ f(n,m)$: there exist positive constants c_1, c_2, n_0, and m_0 such that

$0 \leq c_1 \cdot g(n,m) \leq f(n,m) \leq c_2 \cdot g(n,m)$ for all $n \geq n_0$ and $m \geq m_0 \}$