

CS208 Assignment 2

Author: 王超逸 Chaoyi WANG

SID: 11811014

Content: Answers of q1 and q5 of chapter 2 of the textbook.

2.1

Q: Suppose you have algorithms with the five running times listed below. (Assume these are the exact running times.) How much slower do each of these algorithms get when you (a) double the input size, or (b) increase the input size by one?

- (a) n^2
- (b) n^3
- (c) $100n^2$
- (d) $n \log n$
- (e) 2^n

A:

Running Time	Double the input size	Increase the input size by one
n^2	$4n^2$, 4 times slower.	n^2+2n+1
n^3	$8n^3$, 8 times slower.	n^3+3n^2+3n+1
$100n^2$	$400n^2$, 4 times slower.	$100n^2+200n+100$
$n \log n$	$2n(\log n + \log 2)$, about 2 times slower. If n is small, an additive time cost $2n \log 2$ is more significant.	$(n+1)\log(n+1)$
2^n	4^n , the time cost has been squared.	2^{n+1}

2.5

Q: Assume you have functions f and g such that $f(n)$ is $O(g(n))$. For each of the following statements, decide whether you think it is true or false and give a proof or counterexample.

- (a) $(\log 2)f(n)$ is $O((\log 2)g(n))$.
- (b) $2^{f(n)}$ is $O(2^{g(n)})$
- (c) $f(n)^2$ is $O(g(n)^2)$

A:

$f(n)$ is $O(g(n))$ means that there is a constant c making $f(n) \leq cg(n)$

(a) False.

Let $g(n) = 1$ and $f(n)$ be any positive constant, $(\log_2)g(n) = 0$ while the runtime cannot be $O(0)$.

(b) False.

Let $f(n) = 2n$ and $g(n) = n$, $f(n)$ is $O(g(n))$ but $2^{f(n)} = 4^n$ is not $O(2^n)$.

(c) True.

Since $f(n) \leq cg(n)$ and both $f(n)$ and $g(n)$ are positive, we can infer that $f(n)^2 \leq c^2(g(n)^2)$, making the statement true.