Reid Roberts
01/08/2025

CPSC 6109 Algorithms Analysis and Design

Assignment 01: Analysis of algorithms and Big-oh notation.

Possible points: 60

1. Big-Oh: Determine the Big-Oh of the growth functions below: **20 points**

I.	$5N + 4$
II.	$9N^2 + 8N - 7$
III.	$2 \log N$
IV.	$3N \log 4N$
V.	$6*2^{7N}$
VI.	$3N! + 1/2*N^3$
VII.	42
VIII.	$2*N^3 + 999 * N^2 + 123456789*N$

i.) $O(n)$
ii.) $O(n^2)$
iii) $O(\log n)$
iv.) $O(n \log n)$
v.) $O(2^{7n})$
vi.) $O(n!)$
vii.) $O(1)$
viii.) $O(n^3)$

Stuck? Plug in some numbers to get a better idea.

2. What does the following algorithm do? Analyze (provide detail explanation) its worst-case running time, and express it using "Big-Oh" notation. **20 points**

- k increments until it is not less than n^2
- each operation in the loop runs in $O(1)$ time for n^2 loops
- a gets multiplied by itself n^2 times

Algorithm Foo (a, n):
Input: two integers, a and n
Output: ?
$k \leftarrow 0$
$b \leftarrow 1$
while $k < n^2$ **do**
$k \leftarrow k + 1$
$b \leftarrow b * a$
return b

running time is $O(n^2)$
returns a^{n^2}

3. What does the following algorithm do? Analyze (provide detail explanation) its worst-case running time, and express it using "Big-Oh" notation. **20 points**

- if using floating point division the loop then assume k will never reach 0 and it will be an infinite loop for $n > 0$
- if using integer division, then k will reach 0 in logarithmic time since it is halved each loop
- $k = n^2$

$\log_2(n^2) = 2\log_2(n)$
$= c \cdot \log(n)$
$= O(\log n)$

Algorithm Bar (a, n):
Input: two integers, a and n
Output: ?

k ←n^2
b ←1
c ←a
while k > 0 **do** → $O(\log n)$
k ←k/2 → $O(1)$
c ←c*c $O(1)$
b ←b*c $O(1)$
return b

running time is $O(\log n)$

Submission: Please submit a single PDF containing all the answers.