CMPSC 465 – Spring 2021 — Solutions to Homework 1

Josiah Kim, SID 948500821, `juk483`

January 28, 2021

1. Acknowledgements

(a) I did not work in a group.

(b) I did not consult with any of my group members.

(c) I did not consult any non-class materials.

2. Tribonaci numbers

(a) $R(i) = R(i-1) + R(i-2) + R(i-3) \geq 3^{i/2}$ for all i ≥ 2

BASE CASE: $i = 2$
$R(2) = 3 \geq 3^{2/2}$
$3 \geq 3$
This is true.

INDUCTIVE HYPOTHESIS: For $n > 2$, assume that the claim holds for all $2 \leq i \leq n$.
Assume $R(n)$ is true.
$R(n) = R(n-1) + R(n-2) + R(n-3) \geq 3^{n/2}$

INDUCTIVE STEP: $i = n+1$
$R(n+1) = R((n+1)-1) + R((n+1)-2) + R((n+1)-3) \geq 3^{(n+1)/2}$
$R(n+1) = R(n) + R(n-1) + R(n-2) \geq 3^{(n+1)/2}$ $R(n+1) = R(n) + R(n-1) + R(n-2) \geq (R(n-1) + R(n-2) + R(n-3)) * 3^{1/2}$ (By I.H.)
$R(n+1) = (R(n) + R(n-1) + R(n-2))/(R(n-1) + R(n-2) + R(n-3)) \geq 3^{1/2}$

This means $R(n) \geq 3^{n/2}$ is true for $n \geq 2$.
Therefore, $R(i) \geq 3^{n/2}$ is true for all $i \geq 2$.

(b) SEE BELOW

3. Big Oh Definitions

PROBLEM: Need to prove $O_1(g(n)) \subseteq O_2(g(n))$ AND $O_2(g(n)) \subseteq O_1(g(n))$.

PROOF FOR $O_1(g(n)) \subseteq O_2(g(n))$:
Assume that $c_2 = 10$, $g(n) = n^2$.
This means that $f(n) \leq 10n^2$.
Assume that $c_1 = 10$, $g(n) = n^2$.
This means that $f(n) \leq 10n^2$.
Using the same constant and polynomial parameters, $f(n) \leq 10n^2$ in $O_2(g(n))$ is also part of $O_1(g(n))$.
Therefore, $O_1(g(n)) \subseteq O_2(g(n))$.

PROOF FOR $O_2(g(n)) \subseteq O_1(g(n))$.
Assume that $c_1 = 25$, $g(n) = 3^n$.
This means that $f(n) \leq 25(3^n)$.
Assume $c_2 = 25$, $g(n) = 3^n$.
This means that $f(n) \leq 25(3^n)$.
Using the same constant and polynomial parameters, $f(n) \leq 25(3^n)$ in $O_1(g(n))$ is also part of $O_2(g(n))$.
Therefore, $O_1(g(n)) \subseteq O_2(g(n))$.

CONCLUSION:
It is true that $O_1(g(n)) \subseteq O_2(g(n))$ AND $O_2(g(n)) \subseteq O_1(g(n))$.
Therefore, $O_1(g(n)) = O_2(g(n))$ for all g.

4. Analyze Running Time

(a) The "for" loop will have a time complexity of n.
The nested "while" loop will have a time complexity of n as well.
We will represent the $j + 5$ as 1.
$T(n) = n * 1n$
$T(n) = n^2$
Therefore, $\Theta(n^2)$.

(b) The "for" loop will have a time complexity of n.
The nested "for" loop will also have a time complexity of n.
The arithmetic step, $s + 2$ will be a constant, 1.
$T(n) = n * 1n$
$T(n) = n^2$
Therefore, $\Theta(n^2)$.

(c) There are two simple arithmetic operations ($i \div 2$, $s + 1$) which will each have a time complexity of 1.
The "while" loop will have a time complexity of n.
The nested "for" loop will have a time complexity of n.
$T(n) = 1n * 1n$
$T(n) = n^2$
Therefore, $\Theta(n^2)$.

(d) The "for" loop will have a time complexity of n.
There are two operations which will each have a time complexity of 1.
The nested "while" loop will have a time complexity of n.
$T(n) = 1n * 1n$
$T(n) = n * n$
Therefore, $\Theta(n^2)$.

5. Tighter Analysis

Number, Flips

1, 1
2, 2
3, 1
4, 3
5, 1
6, 2
7, 1
8, 4
9, 1
10, 2
11, 1
12, 3
13, 1
14, 2
15, 1
16, 5
17, 1
18, 2
19, 1
20, 3
21, 1
22, 2
23, 1
24, 3
25, 1
26, 2
27, 1
28, 3
29, 1
30, 2
31, 1

Total flips to 31: 56

Looking at the pattern of bit-flips, every four steps is about seven bit-flips.
$T(n) = ((2^n - 1)/4) * 7 = \Theta(2^n)$
This statement holds true since $((2^n - 1)/4) * 7 \geq 2^n$.