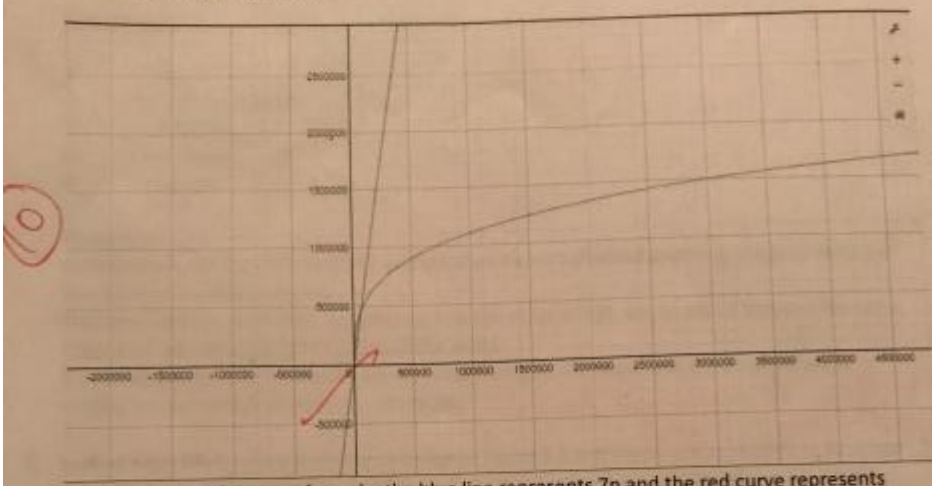
ANALYSIS OF ALGORITHMS – COMP 4040
ASSIGNMENT-1 SOLUTIONS

---

1. CREDITS: Minh Nguyen

Homework 1

1. **Compare Functions:** What is the smallest integer value of $n > 3$ such that an algorithm whose running time is $7n$ runs slower than an algorithm whose running time is $7(\log_2 n)^4$ on the same machine? Justify your answer.



Here is a graph of the two formula, the blue line represents 7n and the red curve represents $7(\text{Log}_2 n)^4$

The smallest integer value whose running time of 7n is slower than of $7(\log_2 n)^4$ is **65537**.

PROOF:

We will be testing 65536 and 65537 for the two formulas

$Y = 7n$
$Y = 7 \times 65{,}535$
$Y = 458{,}752$

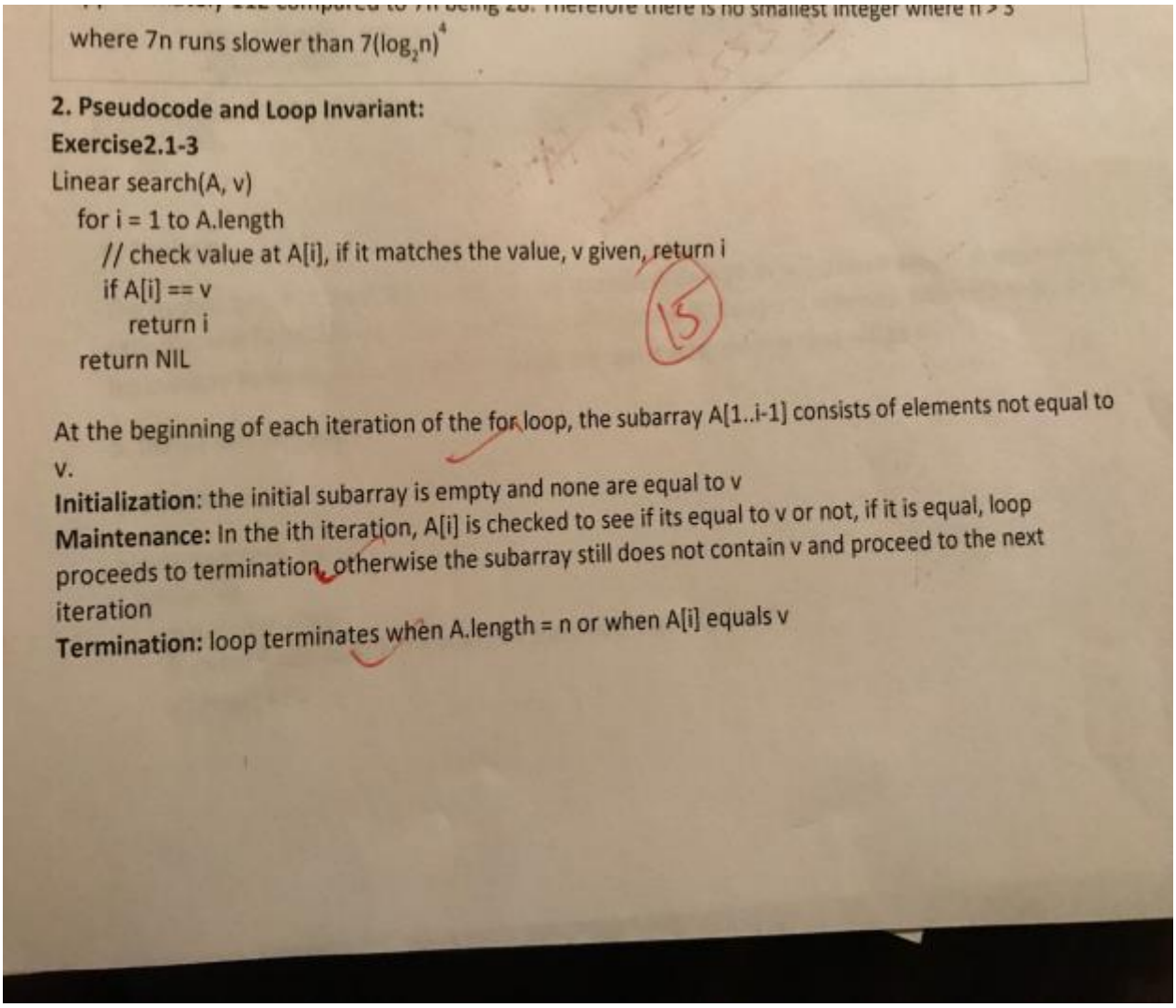
$Y = 7n$
$Y = 7 \times 65{,}537$
$Y = 458{,}759$

$Y = 7(\log_2 n)^4$
$Y = 7(\log_2 65{,}536)^4$
$Y = 458{,}752$

$Y = 7(\log_2 n)^4$
$Y = 7(\log_2 65{,}537)^4$
$Y = 458{,}754$

As you can see, at 65537 is where the log function starts to take less time to calculate and be more optimized.

2. CREDITS: Christopher Pearce



where $7n$ runs slower than $7(\log_2 n)^4$

2. Pseudocode and Loop Invariant:

Exercise2.1-3

```
Linear search(A, v)
   for i = 1 to A.length
      // check value at A[i], if it matches the value, v given, return i
      if A[i] == v
         return i
   return NIL
```

15

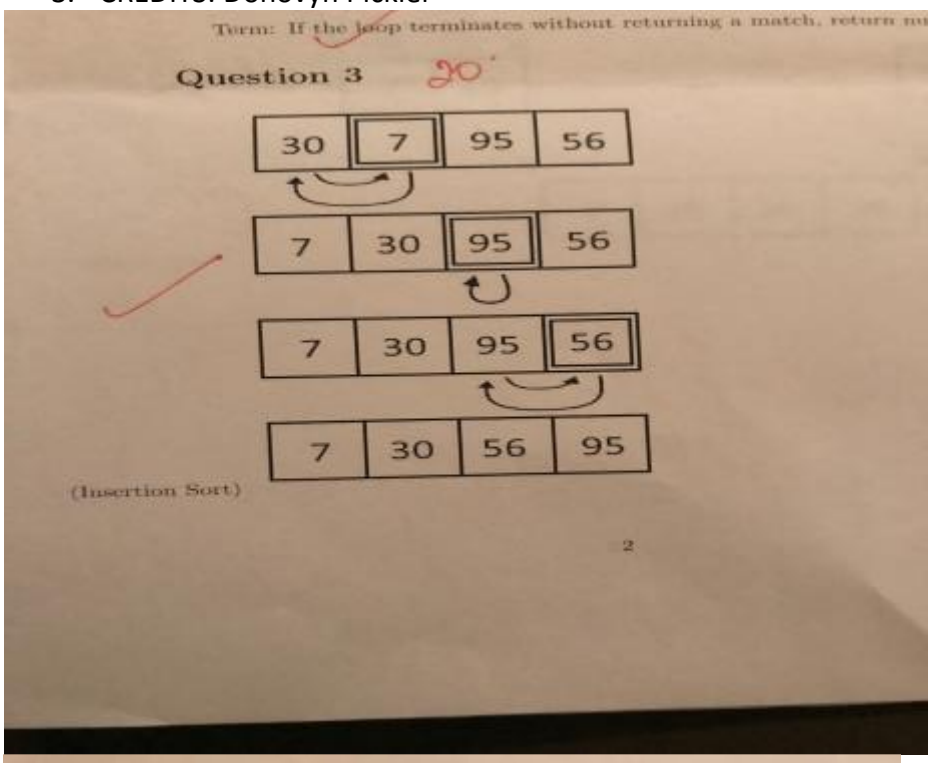
At the beginning of each iteration of the for loop, the subarray A[1..i-1] consists of elements not equal to v.

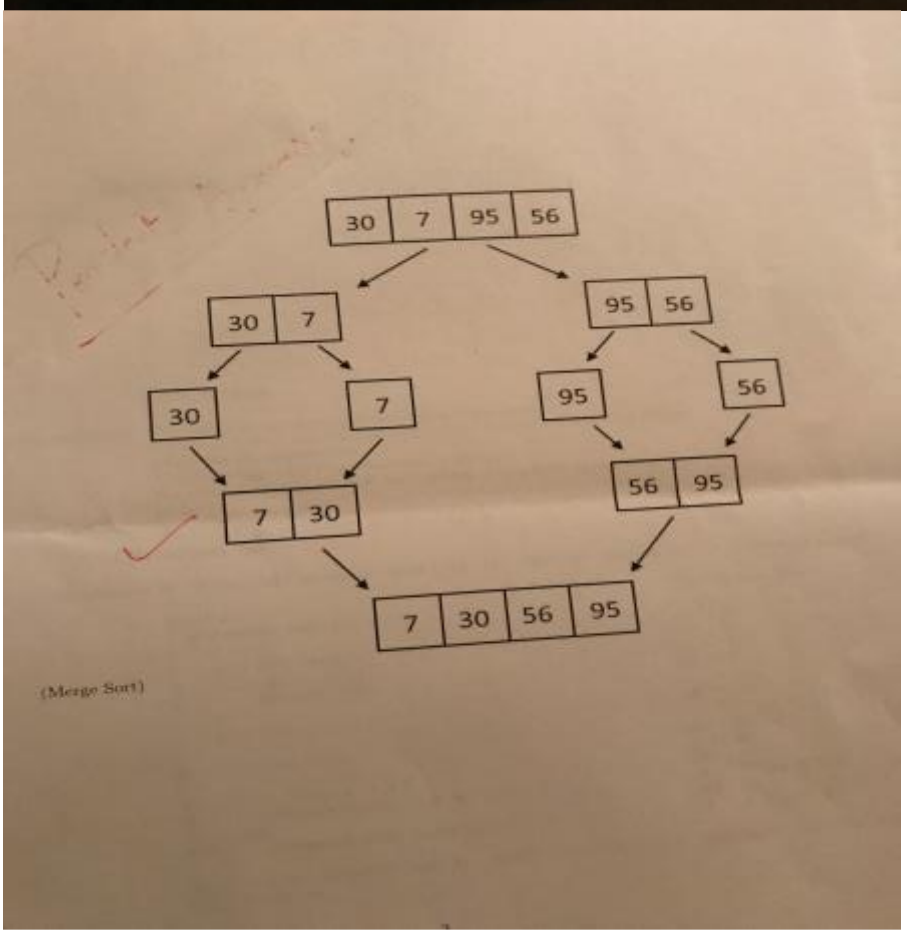
Initialization: the initial subarray is empty and none are equal to v

Maintenance: In the ith iteration, A[i] is checked to see if its equal to v or not, if it is equal, loop proceeds to termination, otherwise the subarray still does not contain v and proceed to the next iteration

Termination: loop terminates when A.length = n or when A[i] equals v

3. CREDITS: Donovyn Pickler





4. CREDITS: Steve Kim

	Mystery (n)	cost	times
1	$if\ n \leq 1$	$c1$	1
2	$return\ 1$	$c2$	1
3	$for\ i = 1\ to\ 5$	$c3$	6
4	$for\ j = 1\ to\ n^2$	$c4$	$5(n^2 + 1)$
5	$print\ "Welcome\ to\ recursion!"$	$c5$	$5(n^2)$
6	$Mystery(n/3)$	$T(n/3)$	1
7	$Mystery(n/3)$	$T(n/3)$	1
8	$Mystery(n/3)$	$T(n/3)$	1

$$T(n) = c1 + c2 + 6c3 + c4 * 5(n^2 + 1) + c5 * 5(n^2) + 3T(n/3) = cn^2 + 3T(n/3)$$

This recursive solution then becomes.....

$$T(n) = cn^2 + \left(\frac{cn^2}{3}\right) + \left(\frac{cn^2}{9}\right) + \left(\frac{cn^2}{27}\right) + \cdots + \left(\frac{cn^2}{3^i}\right)$$

$$\leq cn^2 \sum_{i=0}^{\infty} \frac{1}{3^i}$$

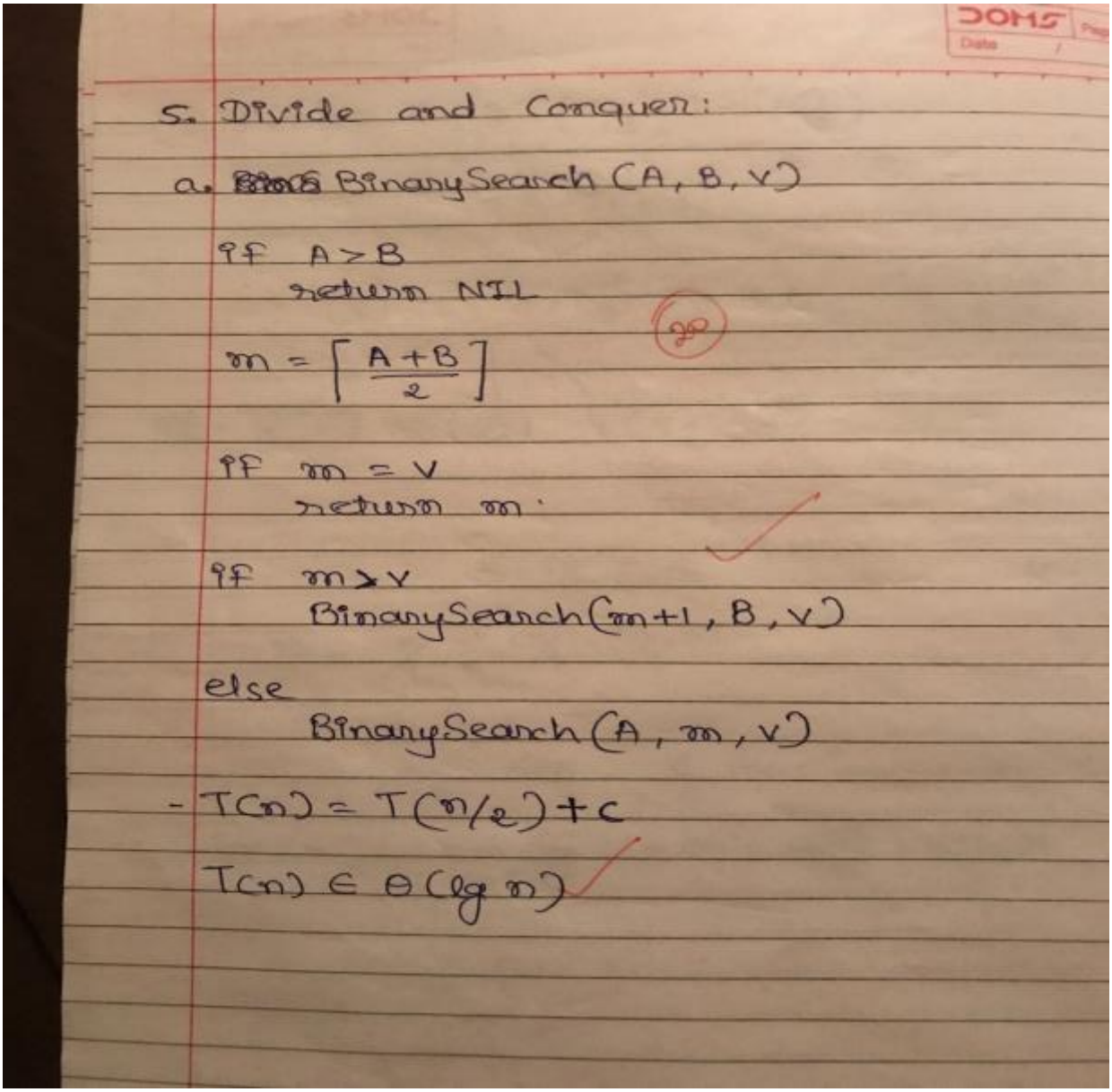
The summation is geometric and converges to 3/2

$$\leq \frac{3}{2} cn^2$$

Thus, the worst-case asymptotic execution time of Mystery is $T(n) = \theta(n^2)$

(Courtesy of Micah Twombly)

5. CREDITS: Gangi Gajjan



Problem 5.b

First, sort the elements in **S**, taking **Θ(n** log **n)** time. Then, for each element **y** in **S**, perform a binary search in **S** for **x-y**. Each binary search takes **O(**log **n)** time, and there are are most **n** of them, and so the time for all the binary searches is **O(n** log **n)**. The overall running time is **Θ(n** log **n)**.